Localization and Change Point Detection using GPS Data

by

Xiaoyu Zhai

A thesis
submitted to Victoria University of Wellington
in fulfilment of the
requirements for the degree of
Master of Science
in Applied Statistics.

Victoria University of Wellington
2013

Abstract

The Global Positioning System (GPS) has become widely used in modern life and most people use GPS to find locations, therefore the accuracy of these locations is very important.

In this thesis, we will use Longitude and Latitude from raw GPS data to estimate the location of a GPS receiver. To improve accuracy of the estimation, we will use two methods to delete outliers in Longitude and Latitude: the Euclidean distance method and the Mahalanobis distance method. We will then use two methods to estimate the location: Maximum Likelihood and Bootstrap method.

The confidence ellipse and the simultaneous confidence intervals are used to construct confidence regions for bivariate data, and we compared the two methods. In this thesis, we also did some simulations to understand the effect of sample size and variance in the linear regression model for AIC and BIC, and use these two criteria to find a best model to fit the multivariate linear regression model with response variables Latitude and Longitude.

This thesis forms part of a larger project to detect land movement, such as that seen in landslides using low cost GPS devices. We therefore consider methods for detecting changes in location over time.

In this thesis, we used converted Longitude, Latitude and Altitude (in meters) from the same GPS data set after deleting outliers as our variables and applied two methods (Hotelling's T^2 chart method and Multivariate exponentially weighted moving average method) to detect changes in location in our data.

Acknowledgments

First of all, I would like to thank my first supervisor Dr. Nokuthaba Sibanda. She played an important role in completion of my thesis. I am lucky to have found a mentor who is understanding and knows how to push me so that I achieve my best. I would also like to thank my second supervisor Dr. Yuichi Hirose. I greatly appreciate his patience and hard work while I was working on my thesis.

I would like to offer my special thanks to my parents who supported me every step of the way to reach the goal.

At last, a thanks to my office mates who also encouraged me and gave me technical support.

Contents

Chapter 1

Introduction

Positioning in the Global Positioning System (GPS) can be performed in either of two ways: point positioning or relative positioning. GPS point positioning employs one GPS receiver that measures the code pseudoranges to determine the user's position instantaneously, as long as four or more satellites are visible to the receiver. The expected horizontal positioning accuracy from the civilian C/A-code receivers has gone down from about 100m (2 drms) when selective availability was on, to about 22m (2 drms) in the absence of selective availability [43]. GPS point positioning is used mainly when relatively low accuracy is acceptable. This includes recreational applications and low accuracy navigation [12].

GPS relative positioning, however, employs two GPS receivers simultaneously, tracking the same satellites. If both receivers track at least four common satellites, a positioning accuracy level of the order of a subcentimeter to a few meters can be obtained [18]. Carrier-phase and/or pseudorange measurements can be used in GPS relative positioning, depending on the accuracy requirements. The former provides the highest possible accuracy. GPS relative positioning is used for high-accuracy applications such as surveying and mapping, geographic information systems (GIS), and precise navigation [12].

GPS pseudorange and carrier-phase measurements are both affected

by several types of random errors and biases (system errors). These errors may be classified into three groups, those originating at the satellites, those originating at the receiver, and those that are due to signal propagation [24].

The errors originating at the satellites include ephemeris or orbital errors, satellite clock errors, and the effect of selective availability. The errors originating at the receiver include receiver clock errors, multipath errors, receiver noise, and antenna phase center variations. The signal propagation errors include the delays of the GPS signal as it passes through the ionospheric and tropospheric layers of the atmosphere. In addition to the effect of these errors, the accuracy of the computed GPS position is also affected by the geometric locations of the GPS satellites as seen by the receiver. The more spread out the satellites are in the sky, the better the obtained accuracy [12].

This thesis forms part of a larger research project to develop a system for detecting land movement using low cost GPS devices. The broader goal of the research project is to use changes in positions of these GPS devices to indicate land movement such as that seen in landslides. Point positioning will be used since a low cost system is required.

In this thesis, a single GPS device will be placed in a fixed location and its position will be estimated from its position readings, many of which will be inaccurate. Methods for detecting changes in locations will then be considered.

1.1 Research Goals

In this thesis, the goals are:

1. Estimate the position of a GPS device from multiple measurements taken sequentially throughout the day.

   We use longitude and latitude in radians to estimate the position of

a GPS device and also use longitude, latitude and altitude converted to meters to estimate the position of a GPS device.

2. Determine whether position measurements are dependent on other factors.

3. Detect changes in position of receiver from position measurement.

There are lots of statistical methods for position estimation.

Bayesian-filter techniques for location estimation is introduced by Fox et al. (2003). This technique provides a powerful statistical tool to help manage measurement uncertainty and perform multi-sensor fusion and identity estimation. It also discusses different belief representations and their properties in the context of location estimation for pervasive computing [14].

Using Linear Discriminant Analysis (LDA) to estimate location is introduced by Zhou et al. (2006). They proposed a selector method with LDA among different kinds of mobile location estimation algorithms and provide a more accurate estimation for location service. The LDA is a statistical method based on the variance analysis and its main idea is to use projection to separate different class data [48].

Another method for mobile location estimation is introduced by Lin and Juang (2003). This estimation is based on differences of signal attenuations in Global System for Mobile Communications (GSM). The advantages of this proposed method is that the method does not need perfect path loss modelling, it also reduces the shadowing impact on location, it has low computational complexity, and it can be applied in existing systems without hardware development [29].

An approach to location estimation is introduced by Roos et al. (2002). This approach is called the statistical modelling approach. They present a location estimation method based on a statistical signal power model and also present encouraging empirical results from simulated experiments

supported by real-world field tests. The advantages of the approach includes certain types of flexibility that presented itself in the present work [39].

A landslide is a type of mass movement that causes damage in many areas. Elastic image registration and the change-unchanged conditional statements procedure is introduced by Shariff et al. (2011) for historical analysis of the land movement in landslide areas. Landslide areas were detected using the amount of pixel movement during a registration process [21].

Kostiuk outlined the use of remote sensing data to detect sea level change in 2002. Remote sensing data is useful to check if there has been significant change in a coastline or to detect various types of environment conditions [27].

Light detecting and ranging (LIDAR) is being used in aerial and terrestrial scanning, offering laser images and measurements that can seemingly strip away vegetation and structures, see changes in landforms as they happen, and create a permanent record of how the landscape looked at a particular moment in time including right after a natural disaster. LIDAR is a good analytic tool to help with detecting land movement [34].

1.2 Thesis Outline

The outline of the rest of this thesis is as follows:

Chapter 2 describes position estimation of a GPS device using Longitude, Latitude and Altitude. It introduces two methods to detect and delete outliers.

Chapter 3 introduces a bootstrap method to estimate the position and the bootstrap simultaneous confidence intervals (SCI) to construct confidence intervals for bivariate data.

Chapter 4 investigates the performance of AIC and BIC in linear regression models and uses AIC and BIC to select the multivariate linear

regression models for the response variables: Longitude and Latitude.

Chapter 5 uses two change point methods to detect any change in position of receiver in our time frame for Longitude, Latitude and Altitude.

Chapter 6 concludes what we did in this thesis and some discussion of the results in this thesis.

Chapter 2

Position Estimation using GPS Data

2.1 Introduction

In this chapter, the aim is to estimate the location of a GPS receiver using Longitude and Latitude, and to construct a confidence region for the location. The method is used for this chapter is assuming that the data that we used follow a normal distribution.

We work on the GPS data set. Each observation of the GPS data set is from a fixed location GPS navigation device that received GPS signals for the purpose of determining the device's current location on Earth.

There are 12 variables and 2913 observations in this data set. The 12 variables are:

- "$NavValid$" is valid navigation.
- "$NavType$" is the GPS position fix type.
- "$GPSWeek$" is GPS week number, weeks since January 6, 1980.
- "$GPSTOW$" is GPS time of week in seconds $\times 10^{-3}$.

- "$Satellites\ used\ for\ solution$".

- "$NumSats$" is the number of satellites used.

- "$Latitude$" is in degrees $(+ = North) \times 10^7$.

- "$Longitude$" is in degrees $(+ = East) \times 10^7$.

- "$Altitude.MSL$" is in meters $\times 10^2$.

- "$Altitude.Ellipsoid$" is in meters $\times 10^2$.

- "$Horizontal\ error$" is in meters $\times 10^2$.

- "$Vertical\ error$" is in meters $\times 10^2$.

The 2913 observations were collected over a 40-hour period. The bivariate data used are Latitude and Longitude from a GPS device, and are estimated from satellites. The new data set used for the project consists of those observations for which NumSats is greater than zero, otherwise an observation is not valid. Removing these observations leaves 2671 observations in the data set.

For clear understanding and presentation, Latitude and Longitude are divided by 10^7, converting them into degrees. GPSTOW is divided by 3600000, converting GPSTOW into hours.

The rescaled Latitude and Longitude are plotted below:

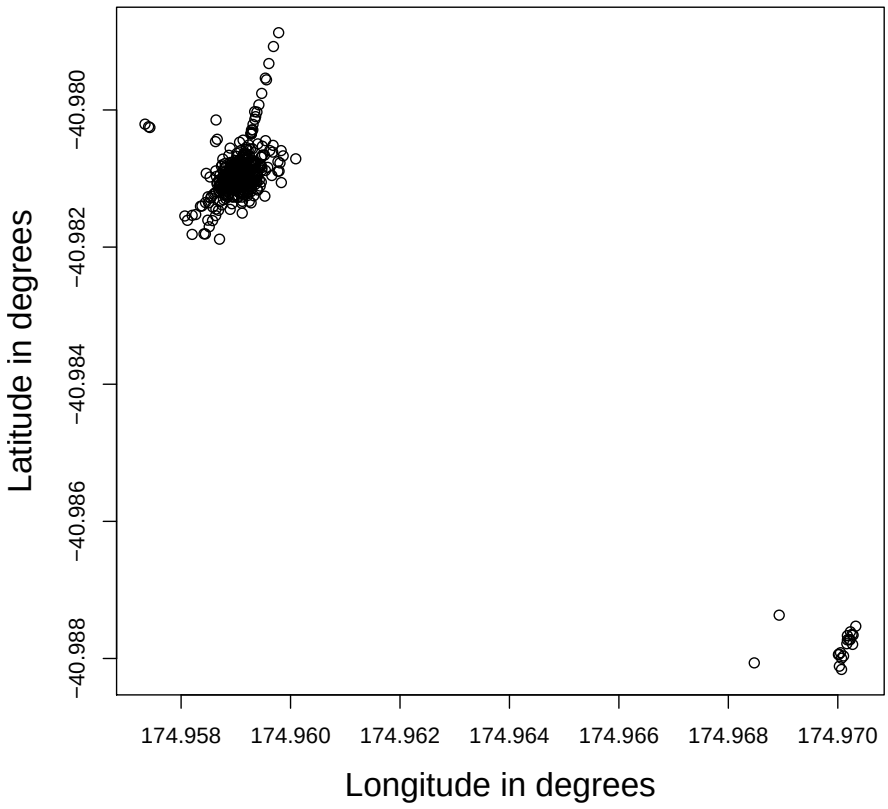


Figure 2.1: Observations of Longitude and Latitude

From this plot, we can clearly see what values of Longitude and Latitude are most common for the positions of the observations. Also we notice there are some outliers. So we need to remove outliers to improve estimation of the position of the GPS navigation device.

The 3D plot below is Latitude and Longitude against GPSTOW.

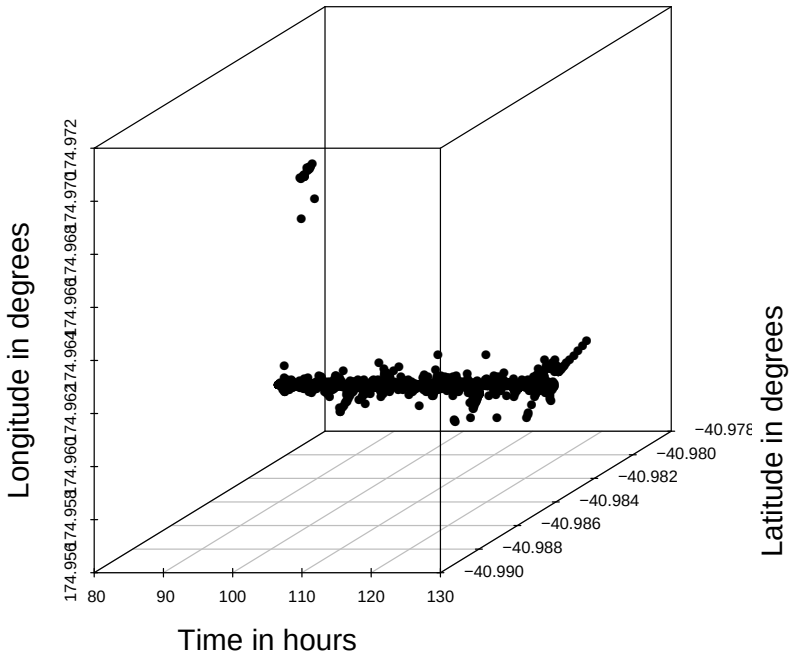


Figure 2.2: 3D plot for Longitude, Latitude and GPSTOW

From the plot, we can see there is a main straight line parallel to the time variable and some points which are further away from the line, these are outliers. We can see the outliers occur close together in time.

2.2 Detecting and deleting outliers

What are outliers?

In statistics, an outlier is an observation that is numerically distant from the rest of the data [3].

In many data analysis tasks a large number of variables are being recorded or sampled. One of the first steps towards obtaining a coherent analysis is the detection of outlying observations. Although outliers are often considered as errors or noise, they may carry important information. Detected outliers are candidates for abnormal data that may lead to model misspecification, biased parameter estimation and incorrect results. It is therefore important to identify them prior to modelling and analysis [5].

Outliers are best detected visually whenever this is possible. For a single random variable which is one dimensional, the outliers are the observations that are far from the others [20].

Outlier detection is a very important process in data analysis.

There are three fundamental approaches to the problem of outlier detection. One of the approaches concerns the detection of outliers with no prior knowledge of the data [17]. This type of outliers is the type that we are dealing with.

There are four criteria introduced to determine outliers: PanTa criterion, Grubbs criterion, Dixon criterion and Chauvenet criterion. These criteria require normally distributed data, otherwise the reliability of the judgement will be affected [16].

There are many ways to detect and delete outliers. Probably one of the simplest statistical outliers detection techniques is described here: Laurikkala et al. (2003) uses information box plots to pinpoint outliers in both univariate and multivariate data sets. For multivariate data sets the authors note that there are no unambiguous total orderings but recommend using the reduced sub-ordering based on the generalised distance metric using the Mahalanobis distance measure. The Mahalanobis distance measure includes inter-attribute dependencies so the system can compare attribute combinations [17].

The Mahalanobis distance, given by

$$\sqrt{(\boldsymbol{x}_i - \boldsymbol{\mu})^T \mathbf{C}^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu})},$$

calculates the distance from a point $\boldsymbol{x}_i$ to the centroid ($\boldsymbol{\mu}$) defined by cor-

related attributes with the covariance matrix $(\mathbf{C})$.

An outlier determining method is applied for multivariate data, by using reduced sub-ordering of each multivariate observation $\mathbf{x}_i$ to a scalar r_i as shown in the equation below. These scalars can then be ordered as univariate observations [8].

$$r_i^2 = (\mathbf{x}_i - \mathbf{x}_0)'\Gamma^{-1}(\mathbf{x}_i - \mathbf{x}_0)$$

where $\mathbf{x}_0$ is a reference point (or an origin) and Γ^{-1} weights variables inversely to their scatter.

Two classes of models are proposed for the detection of outliers by Kitagawa (1979) [23]. The best approximating model is obtained by minimizing AIC, so that we no longer need to determine type and number of outliers and hypothesis test.

Zhao (2003) [47] analyses and compares some methods for judging the outliers in univariate and multivariate cases. To detect outliers in the normal distribution, there are many methods: (1) Grubbs test, (2) t test, (3) Dixon test, (4) Nair test , (5) Skewness-Kurtosis test. Zhao (2003) states these test methods have limitations in some situations. The test methods 1,4,5 all use the mean of the data to estimate the mean parameter, so the stability is poor. The test methods 1 and 3 can not test many outliers. The test methods 2 and 3 are only used for small sample sizes. The test method 4 can only be used if the variance is known. So it states that using robust estimation tests of the parameters for the population is a good way to judge outliers.

Using Mahalanobis distance method can not always detect outliers, because it is based on the mean and variance-covariance matrix of the data which are affected by the outliers. Computing distance by robust methods is a better way to detect outliers in multivariate dimensions [37].

In our example, we want to delete outliers from our bivariate data. From Figure 2.1, we can see some outliers in the bottom right corner from the main points group and some outliers on top and around the main

points group. There are two distances we will use to delete these outliers, one is the Euclidean distance and the other one is the Mahalanobis distance.

In the Euclidean distance and Mahalanobis distance, the steps to delete outliers are as follows:

1. Let Longitude be denoted by X and Latitude be denoted by Y. Compute the means $\bar{X}$ and $\bar{Y}$ for X and Y respectively.

2. Let X_i be each observation for X and Y_i be each observation for Y, where $i = 1, ..., n$. Compute distance between the means $(\bar{X}, \bar{Y})$ and each observation (X_i, Y_i).

   - for Euclidean distance:

   $$d_i = \sqrt{(X_i - \bar{X})^2 + (Y_i - \bar{Y})^2}$$

   - for Mahalanobis distance:

   $$d_i = \sqrt{\left[\begin{pmatrix} X_i \\ Y_i \end{pmatrix} - \begin{pmatrix} \bar{X} \\ \bar{Y} \end{pmatrix}\right]^T \hat{\text{cov}}(X, Y)^{-1} \left[\begin{pmatrix} X_i \\ Y_i \end{pmatrix} - \begin{pmatrix} \bar{X} \\ \bar{Y} \end{pmatrix}\right]}$$

3. Sort d_i from smallest to largest and delete the largest 1% .

4. Repeat these above steps k times and stop when

   $$\sqrt{(\bar{X}_k - \bar{X}_{k-1})^2 + (\bar{Y}_k - \bar{Y}_{k-1})^2} < 1 \times 10^{-7}.$$

5. Use the data at $(k-1)$th step as the new data set for estimation.

In step 4, 1×10^{-7} is chosen after trying different stopping numbers.

The result of deleting outliers can be clearly seen from the plots below. Each plot is combined with part of the data before deleting outliers (excluding observations in the bottom right corner) and the data after deleting outliers. In Figure 2.3, the red region is the data set after deleting

outliers using the Euclidean distance method. In Figure 2.4, the red region is the data after deleting outliers using the Mahalanobis distance method.

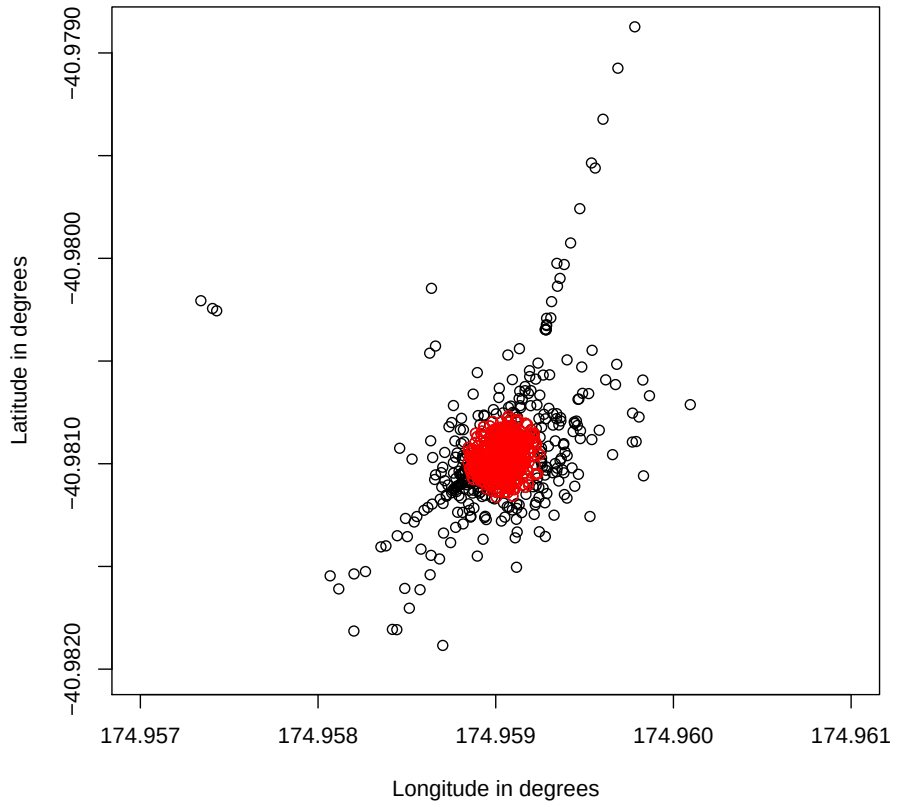


Figure 2.3: Combined plot with data after deleting outliers by the Euclidean distance method

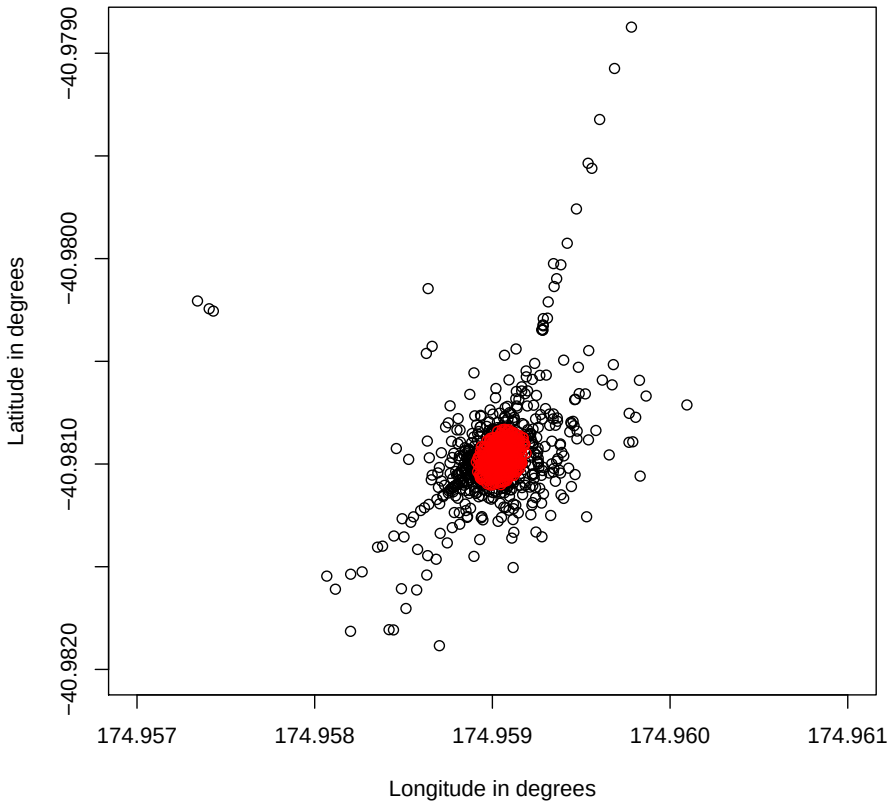


Figure 2.4: Combined plot with data after deleting outliers by the Mahalanobis distance method

From Figure 2.3 and Figure 2.4, we can see that both methods work well for deleting outliers. The data set left after using the Mahalanobis distance method is smaller than the data set left after using the Euclidean distance method. This can be seen as the red region is smaller in Figure 2.4 than in Figure 2.3.

Figure 2.5 and Figure 2.6 shows the difference in means for X and Y

between deletion step k and k-1 calculated as:

$$\Delta^2_{mean} = (\bar{X}_{k-1} - \bar{X}_k)^2 + (\bar{Y}_{k-1} - \bar{Y}_k)^2$$

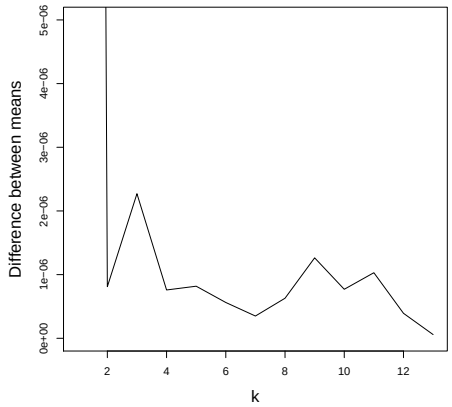


Figure 2.5: Difference of means between steps k and $k-1$ using Euclidean distance method

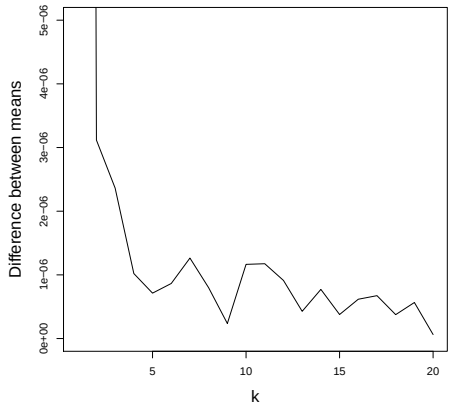


Figure 2.6: Difference of means between steps k and $k-1$ using Mahalanobis distance method

From the above two figures, we can see that there is a big difference between Step 1 and Step 2 in means for both methods, that is because of

the outliers that are very far away from the main observations (clearly seen in the previous figure 2.1). After deleting those outliers in the first step the difference in means between each step for both methods is not steady, but the trend in the difference in means gets smaller.

Figure 2.7 and Figure 2.8 show $\mathrm{Var}(X) + \mathrm{Var}(Y)$ at each deletion step k calculated as:

$$\sigma_k^2 = \sigma_{x_k}^2 + \sigma_{y_k}^2$$

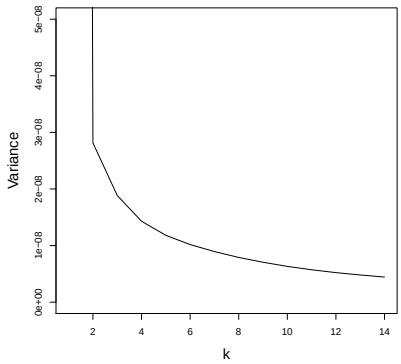


Figure 2.7: Values of $\mathrm{Var}(X) + \mathrm{Var}(Y)$ after each Euclidean deletion step k

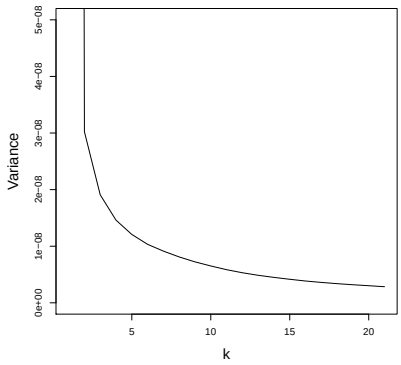


Figure 2.8: Values of $\mathrm{Var}(X) + \mathrm{Var}(Y)$ after each Mahalanobis deletion step k

From the above two figures, we can see that there is a big difference between Step 1 and Step 2 in variance for both methods, it is because of the outliers are far away from the main observations. After deleting those outliers in the first step, the difference in variance at each step for both methods gets smaller and smaller.

From the R output, there are 2362 observations left after step $k = 13$ when the Euclidean distance method is used. The sample mean and variance-covariance matrix are given by:

$$\begin{pmatrix} \bar{X} \\ \bar{Y} \end{pmatrix} = \begin{pmatrix} -40.98097° \\ 174.959° \end{pmatrix}$$

and

$$S = \begin{pmatrix} 2.224118 \times 10^{-9} & 4.476475 \times 10^{-10} \\ 4.476475 \times 10^{-10} & 2.584478 \times 10^{-9} \end{pmatrix}.$$

There are 2198 observations left after step $k = 20$ when the Mahalanobis distance method is used. The sample mean and variance-covariance matrix are given by:

$$\begin{pmatrix} \bar{X} \\ \bar{Y} \end{pmatrix} = \begin{pmatrix} -40.98097° \\ 174.959° \end{pmatrix}$$

and

$$S = \begin{pmatrix} 1.465203 \times 10^{-9} & 3.940516 \times 10^{-10} \\ 3.940516 \times 10^{-10} & 1.558219 \times 10^{-9} \end{pmatrix}.$$

From the two sample means, we can see that there is no difference in means for the two methods.

2.3 Estimation of mean and variance-covariance matrix

In this project, we assume the distribution for the bivariate data is normal.

The density function of the distribution for a random vector (X, Y) is:

$$f(X,Y) = \frac{1}{(2\pi)|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}\left[\begin{pmatrix} X \\ Y \end{pmatrix} - \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}\right]^T \Sigma^{-1} \left[\begin{pmatrix} X \\ Y \end{pmatrix} - \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}\right]\right)$$

The likelihood joint density function for $f(X, Y)$ will be

$$L = \prod_{i=1}^{n} \left\{ \frac{1}{(2\pi)|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}\left[\begin{pmatrix} X_i \\ Y_i \end{pmatrix} - \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}\right]^T \Sigma^{-1} \left[\begin{pmatrix} X_i \\ Y_i \end{pmatrix} - \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}\right]\right)\right\}$$

The log likelihood function for $f(X, Y)$ will be

$$\ln(L) = -n\ln(2\pi) - \frac{n}{2}\ln|\Sigma| - \frac{1}{2}\sum_{i=1}^{n}\left(\left[\begin{pmatrix} X_i \\ Y_i \end{pmatrix} - \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}\right]^T \Sigma^{-1} \left[\begin{pmatrix} X_i \\ Y_i \end{pmatrix} - \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}\right]\right),$$

where μ_1 is the mean of X, μ_2 is the mean of Y, n is the number of observations and Σ is the covariance matrix denoted by

$$\Sigma = \begin{pmatrix} \sigma_{XX} & \sigma_{XY} \\ \sigma_{XY} & \sigma_{YY} \end{pmatrix}.$$

Maximum-likelihood estimation (MLE) is a method of estimating the parameters of a statistical model. In general, for a fixed set of data and a underlying statistical model, the method of maximum likelihood selects values of the model parameters that gives the observed data the greatest probability (i.e., parameters that maximize the likelihood function). Maximum-likelihood estimation gives a unified approach to estimation, which is well-defined in the case of the normal distribution and many other problems.

The maximum likelihood method is used to estimate the mean and the variance covariance matrix for the observations after deleting outliers.

1. Maximum likelihood estimation of the mean $\mu = (\mu_1, \mu_2)$.

Maximizing the log-likelihood over μ is equivalent to minimizing the following function over μ:

$$L(\mu) = \sum_{i=1}^{n}\left(\left[\begin{pmatrix} X_i \\ Y_i \end{pmatrix} - \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}\right] \Sigma^{-1} \left[\begin{pmatrix} X_i \\ Y_i \end{pmatrix} - \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}\right]^T\right)$$

Its derivative with respect to $\mu = \binom{\mu_1}{\mu_2}$ is

$$\frac{\partial Ł}{\partial \mu} = \sum_{i=1}^{n} \left(\Sigma^{-1} \left[\binom{X_i}{Y_i} - \binom{\mu_1}{\mu_2} \right]^T \right)$$

Setting the derivative equal to 0, we have

$$\hat{\mu} = \binom{\hat{\mu}_1}{\hat{\mu}_2} = \frac{1}{n} \sum_{i=1}^{n} \binom{X_i}{Y_i} = \binom{\bar{X}}{\bar{Y}}$$

2. Maximum likelihood estimation of the variance covariance matrix.

The maximum likelihood estimators of Σ is [20]:

$$\hat{\Sigma} = \frac{1}{n} \sum_{i=1}^{n} \left[\binom{X_i}{Y_i} - \binom{\bar{X}}{\bar{Y}} \right] \left[\binom{X_i}{Y_i} - \binom{\bar{X}}{\bar{Y}} \right]^T = \frac{(n-1)}{n} \mathrm{S}$$

where $\mathrm{S} = \frac{1}{n-1} \sum_{i=1}^{n} \left[\binom{X_i}{Y_i} - \binom{\bar{X}}{\bar{Y}} \right] \left[\binom{X_i}{Y_i} - \binom{\bar{X}}{\bar{Y}} \right]^T$.

The MLE estimates for μ and Σ of longitude and latitude are the same as the sample estimates in section 2.2. That is because our sample size is large.

2.4 Confidence region

A $(1-\alpha)100\%$ confidence region for the mean of a 2-dimensional normal distribution and sample size n is given by the set of $\mu \in \mathbf{R}^2$ that satisfies the inequality:

$$n \left[\binom{\bar{X}}{\bar{Y}} - \binom{\mu_1}{\mu_2} \right] \mathrm{S}^{-1} \left[\binom{\bar{X}}{\bar{Y}} - \binom{\mu_1}{\mu_2} \right]^T \leq \frac{2(n-1)}{(n-2)} F_{2,n-2}(\alpha) \qquad (2.1)$$

The inequality defines a 2-dimensional ellipsoid centred at $(\bar{X}, \bar{Y})$ [20].

To construct a confidence ellipse, we need to get the eigenvalue and eigenvector pairs for S denoted by $\lambda_1, \mathbf{e}_1'$ and $\lambda_2, \mathbf{e}_2'$. The eigenvectors $\mathbf{e}_1', \mathbf{e}_2'$ are used to make the directions for axes of the ellipse through the means

$(\bar{X}, \bar{Y})$. The half-lengths of the major and minor axes of the ellipse are given by [20]

$$\sqrt{\lambda_1}\sqrt{\frac{2(n-1)}{(n-2)}F_{2,n-2}(\alpha)}$$

and

$$\sqrt{\lambda_2}\sqrt{\frac{2(n-1)}{(n-2)}F_{2,n-2}(\alpha)}.$$

An indication of the elongation of the confidence ellipse is provided by the ratio of the lengths of the major and minor axes. This ratio is $\frac{\sqrt{\lambda_1}}{\sqrt{\lambda_2}}$.

The "ellipse" function from the package 'mixtools' in R implements the method for constructing confidence ellipses in Johnson and Wichern. It is used to plot the 95% confidence region for X and Y after deleting outliers. "ellipse" is used to draw a two-dimensional ellipse that traces a bivariate normal density contour for a given mean vector, covariance matrix, and probability content.

The Euclidean distance method and 95% confidence region plot is below:

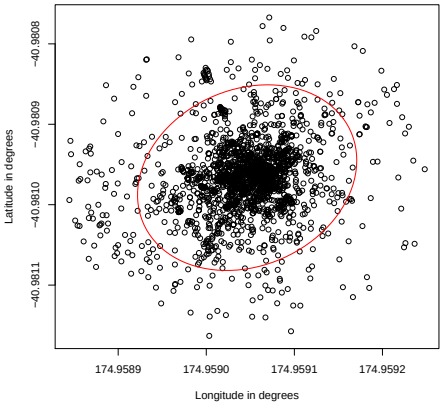


Figure 2.9: 95% confidence ellipse after Euclidean distance method deleting outliers

The Mahalanobis distance method and 95% confidence region plot is below:

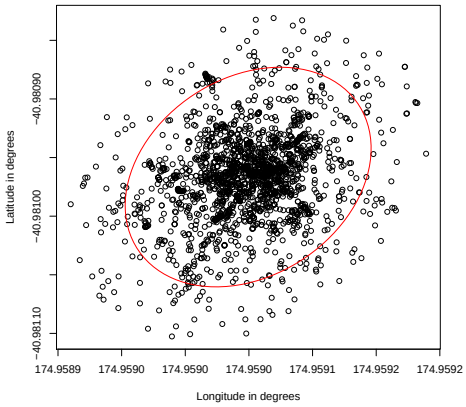


Figure 2.10: 95% confidence ellipse after Mahalanobis distance method deleting outliers

2.5 Position estimation in meters

Since Longitude, Latitude and Altitude can be converted to meters and in some situations need to be in meters, we convert them in this section. The longitude, latitude and altitude from original data set are converted to earth-centred, earth-fixed (ECEF) Cartesian. ECEF also known as ECR (Earth Centred Rotational) is a Cartesian coordinate system, and is sometimes known as the ”conventional terrestrial” system. It represents position as an X, Y, and Z coordinate. The point (0,0,0) is defined as the centre of mass of the Earth, hence the name Earth-Centred. Its axes are aligned with the International Reference Pole (IRP) and International Reference Meridian (IRM) that are fixed with respect to the surface of the Earth, hence the name Earth-Fixed. Let Latitude be denoted by X in meters, Longitude be denoted by Y in meters and Altitude be denoted by Z

in meters. The formula for conversion from degree to meters is below:

$$X = (N + alt) \times \cos(lat) \times \cos(lon)$$
$$Y = (N + alt) \times \cos(lat) \times \sin(lon)$$
$$Z = ((1 - e^2) \times N + alt) \times \sin(lat)$$

where alt is height above WGS84 ellipsoid in meters, lat is Latitude in radians, lon is Longitude in radians, $a = 6378137$, $e = 8.1819190842622e - 2$, $N = a/\sqrt{1 - e^2 \times \sin(lat)^2}$ is prime vertical radius of curvature. WGS84 stands for World Geodetic System of 1984. It is a 3-D, Earth-centered official GPS reference system. It is used in mapping, charting, navigation and so on [32].

After converting, the two distance methods are used to delete outliers as in the previous section 1.2.1, i.e. the Euclidean distance method and the Mahalanobis distance method. The difference in this section is that three variables are used instead of two variables. The three variables are converted longitude, converted latitude and converted altitude, they are denoted by X, Y and Z. The steps to delete outliers are below:

1. Compute the means $\bar{X}$, $\bar{Y}$ and $\bar{Z}$ for longitude, latitude and altitude respectively.

2. Let X_i be each observation for longitude, Y_i be each observation for latitude and Z_i be each observation for altitude, where i is from 1 to n. Compute the distance between the mean $(\bar{X}, \bar{Y}, \bar{Z})$ and each observation (X_i, Y_i, Z_i).

   - for Euclidean distance
   $$d_i = \sqrt{(X_i - \bar{\mathrm{X}})^2 + (Y_i - \bar{\mathrm{Y}})^2 + (Z_i - \bar{\mathrm{Z}})^2}$$
   - for Mahalanobis distance
   $$d_i = \sqrt{\left[\begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix} - \begin{pmatrix} \bar{X} \\ \bar{Y} \\ \bar{Z} \end{pmatrix}\right]^T \mathrm{cov}(X, Y, Z)^{-1} \left[\begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix} - \begin{pmatrix} \bar{X} \\ \bar{Y} \\ \bar{Z} \end{pmatrix}\right]}$$

3. Sort d_i from smallest to largest and delete the largest 1% .

4. Repeat above steps k times and stop when

$$\sqrt{(\bar{X}_k - \bar{X}_{k-1})^2 + (\bar{Y}_k - \bar{Y}_{k-1})^2 + (\bar{Z}_k - \bar{Z}_{k-1})^2} < 0.001$$

5. Use the data at $(k-1)$th step as the new data set for estimation.

Note that in step 4 we choose < 0.001, this is based on after trying different stopping values.

The 3D plot for the three variables of the observations before deleting outliers is :

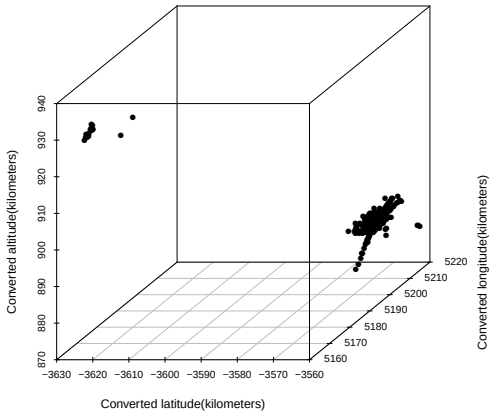


Figure 2.11: 3D plot for the three variables of the observations

The 3D plot for the three variables of the observations after deleting outliers by Euclidean distance method is:

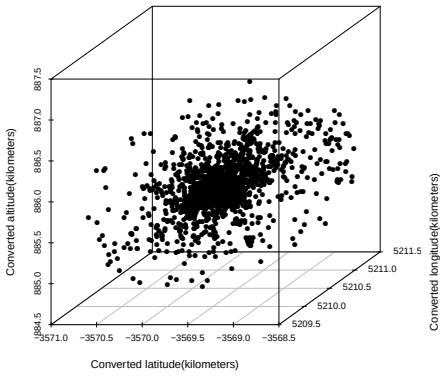


Figure 2.12: 3D plot for the three variables of the observations after deleting outliers by Euclidean distance method

The 3D plot for the three variables of the observations after deleting outliers by Mahalanobis distance method is:

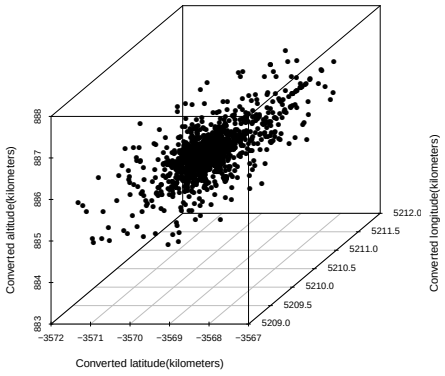


Figure 2.13: 3D plot for the three variables of the observations after deleting outliers by Mahalanobis distance method

There are 2362 observations left after Euclidean distance method delet-

ing outliers and the mean is $(\bar{X}, \bar{Y}, \bar{Z}) = (-3569.6589, 5210.4389, 885.7719)$ in kilometres. There are 2486 observations left after Mahalanobis distance method deleting outliers and the mean is $(\bar{X}, \bar{Y}, \bar{Z}) = (-3569.6371, 5210.4498, 885.7902)$ in kilometres.

The same formulae are used as in the previous section 2.4 to construct a 95% confidence region in a 3 dimension plot.

The 3D plot for the three variables of the observations after deleting outliers by Euclidean distance method with 95% confidence region is:

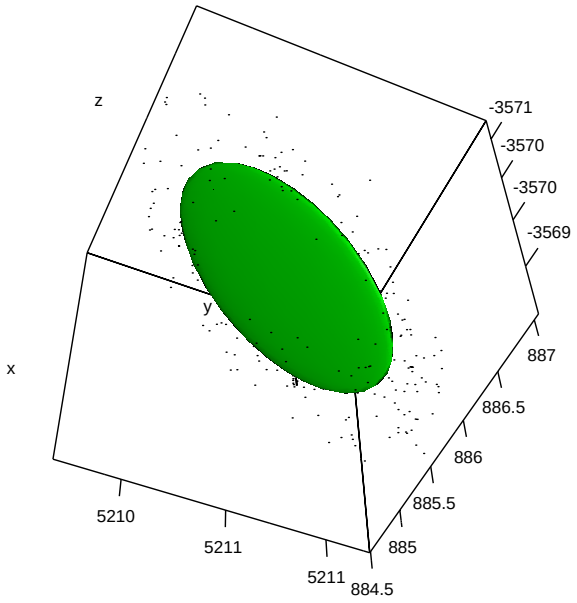


Figure 2.14: 3D plot for Longitude, latitude and altitude with confidence ellipse

The 3D plot for the three variables of the observations after deleting outliers by Mahalanobis distance method with 95% confidence region is:

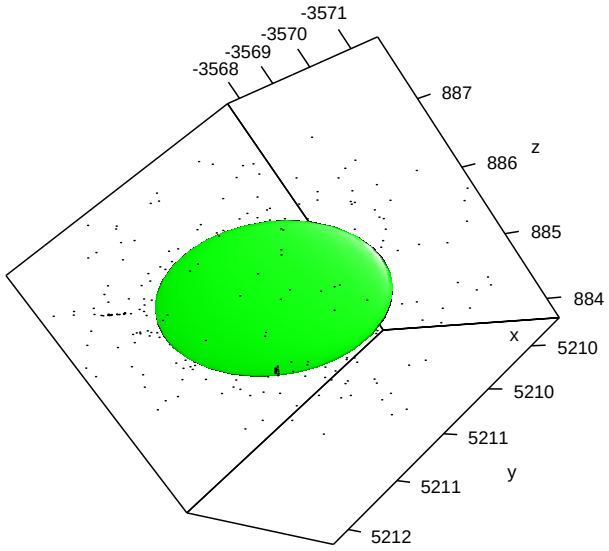


Figure 2.15: 3D plot for Longitude, latitude and altitude with confidence ellipse

2.6 Simulations about outliers

The Euclidean distance method and the Mahalanobis distance method both work well for deleting outliers in GPS data. We want to do some more simulations to test these two methods in three situations and to test which method is better for deleting outliers. These three situations are: when outliers are far away from the data points, when outliers are near the data points, and when outliers are inside the data points.

9800 realisations of bivariate normal data are simulated using R with mean $(\mu_1, \mu_2) = (5, 10)$ and variance-covariance matrix $\mathbf{\Sigma} = \begin{pmatrix} 3 & 1 \\ 1 & 4 \end{pmatrix}$, and is combined with 200 outliers around the data with the same variance-

covariance matrix and they are also normally distributed.
The three situations for outliers are:

1. The outliers are far away from the data points. The mean for the first 100 outliers is chosen to be $(0, 0)$, and the rest 100 outliers is chosen to be $(10, 0)$.

2. The outliers are near the data points. The mean for the first 100 outliers is chosen to be $(-2, 10)$, and the rest 100 outliers is chosen to be $(12, 10)$.

3. The outliers are inside the data points. The mean for the first 100 outliers is chosen to be $(2, 7)$, and the rest 100 outliers is chosen to be $(8, 7)$.

Then the two methods are used to delete outliers: the Euclidean distance method and the Mahalanobis distance method. The steps for deleting outliers are similar to the previous section, the only difference is Step 4: we stop when the difference of the means between each step is less than 0.001. The green dots and red dots are outliers.
The plots of simulated data and outliers are below:

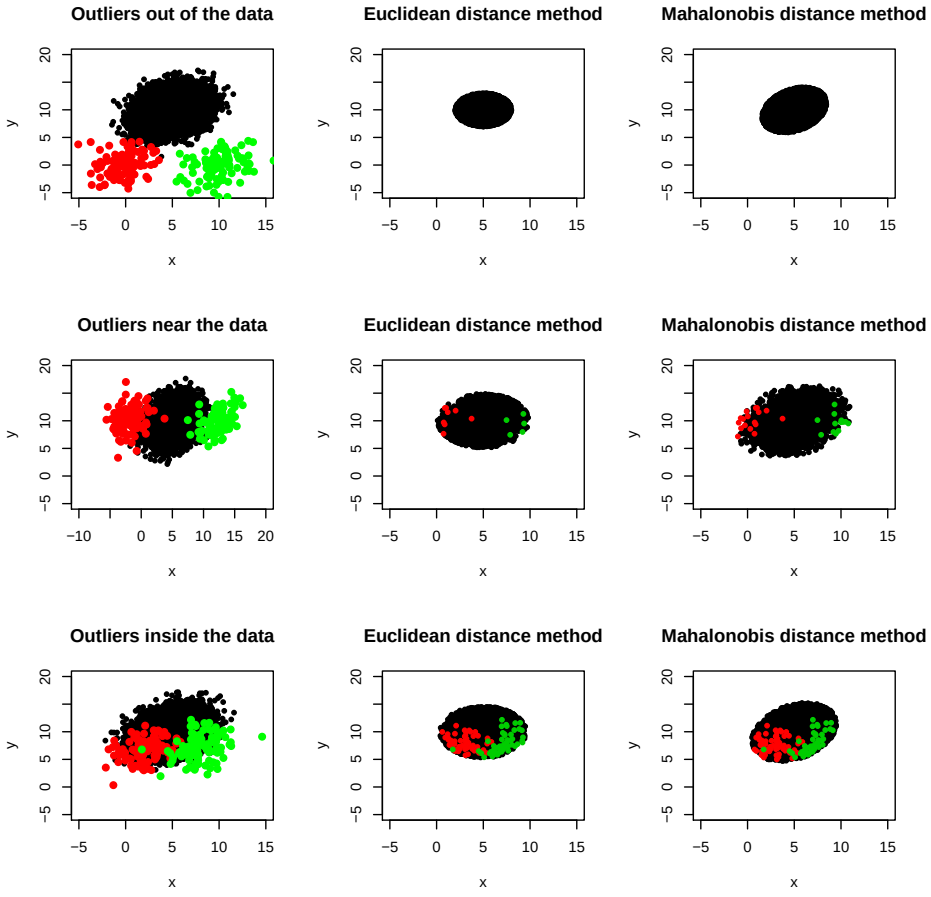


Figure 2.16: Outliers in three situations and removing outliers using two methods

From the plots, we can see that we can easily remove outliers if the outliers are far away from the data, and that it is hard to remove outliers when the outliers are inside the data. This situation happen in our GPS data, that some outliers are inside the GPS data. Therefore we can not remove them and they will influence our accuracy of estimation.

From the R output, we see that it takes more steps to delete outliers using the Euclidean distance method than using the Mahalanobis method.

The Euclidean distance method does not use the covariance matrix for the bivariate data, so the correlation for the bivariate data is 0. We can also see the shape of data after using the Euclidean distance method deleting outliers are upright from the plots. By using the Mahalanobis distance method, we get the similar covariance matrix as the covariance matrix for the data. The means after the two methods are used are not different. Overall, we conclude that the Mahalanobis distance method works better.

2.7 Summary

In this chapter, we worked on the GPS data set, and estimated the position for the GPS navigation device. We create two methods to delete outliers in a bivariate data set, one is the Euclidean distance method and the other one is the Mahalanobis distance method. Then we construct a confidence region for the bivariate data set. The two methods are tested by simulating data in R, and they work well. When we convert degrees to meters for the GPS data set, we get similar results. For an investigative study, we could do some more work for detecting the difficult outliers that may lie inside the data. In this chapter, we assumed the data is bivariate normally distributed, but they are unlikely to be because they are spatial. In the next chapter, we will test for normality and apply the non-parameter method for our data.

Chapter 3

Non-parametric estimation

3.1 Test of normality of the observations

In the previous chapter, the estimation method was based on the assumption of normality of longitude and latitude. In this section, we test whether this assumption is met and investigate a non-parametric approach to the estimation process after removal of the outliers.

The 'mvnormtest' package in R is used to test the normality of observations for bivariate data with X(Longitude) and Y(Latitude) in degrees. This package performs the Shapiro-Wilk test for multivariate normality. The test statistic is [44]:

$$W^* = \frac{1}{2}\sum_{p=1}^{2} W_p$$

where W_p is Shapiro-Wilk's statistic evaluated on the pth variable, where in this case $p = 1, 2$. The formula for W_p is [42]:

$$W_p = \frac{(\sum_{i=1}^{n} a_i x_{(i)})^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

where $x_{(i)}$ is the ith smallest number in the observations, $\bar{x}$ is the observations mean, a_i is constant and given by $(a_1, ..., a_n) = \frac{m^T V^{-1}}{[(m^T V^{-1})(V^{-1} m)]^{1/2}}$ where $m = (m_1, ..., m_n)^T$ are the expected values of the order statistics of

independent and identically distributed random variables sampled from the standard normal distribution, V is the covariance matrix of those order statistics, and n is the number of observations.

W^* is normally distributed with the following approximations [44]:

$$\mu^* = E(W^*) = E\left(\frac{1}{2}\sum_{p=1}^{2} W_p\right) = E(W) = 1 - e^{\mu_n + \sigma_n^2/2}$$

and

$$\sigma^{2*} = \mathrm{Var}(W^*) = \mathrm{Var}\left(\frac{1}{2}\sum_{p=1}^{2} W_p\right) = \frac{1}{2}\mathrm{Var}(W) = \frac{1}{2}[e^{2(\mu_n+\sigma_n^2)} - e^{2\mu_n+\sigma_n^2}].$$

Where W is the Shapiro-Wilk's test, μ_n is the observation mean, and σ_n^2 is the observation variance.

After the test, the result is $p - value < 2.2 \times 10^{-6}$, it means X and Y are not normally distributed.

3.2 Introduction of bootstrap method

After the Second World War, widespread use of computers led to a significant development in mathematical statistics. This development did not only solve difficult computation problems of the past, but also developed new statistical theories that used the new powerful computing capacity efficiently. The bootstrap method is one of the statistical theories that interested statisticians. Since 1980, statisticians have done a lot of research work in the application and theoretical basis of bootstrap method.

The bootstrap is called a re-sampling procedure which is used for re-sampling from the original data set [9].

There are two important problems in applied statistics. One is how to to determine an estimator for a particular parameter of interest. The other one is how to evaluate the accuracy of the estimator using the standard error and confidence interval [9].

From the time that Efron presented the bootstrap method in 1979 until now, the method has been greatly developed and widely used in a variety of statistical fields. As the bootstrap is quite comprehensive, it has been applied to a wider range of problems, such as logistic regression, cluster analysis, non-linear regression, time series analysis, complex surveys and other finite population problems rather than just estimation of standard error and confidence interval [9].

It is not just bound to the world of statistics, but has also been successfully applied in various fields that include accounting, ecology, biology and so on [9].

The non-parametric bootstrap is defined as follows: Given a sample of n independent identically distributed random vectors $\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_n$ and a real-valued estimator $\theta(\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_n)$ (denoted by $\hat{\theta}$) of the distribution parameter θ, a procedure (the bootstrap) to assess the accuracy of $\hat{\theta}$ is defined in terms of the empirical distribution function F_n. This empirical distribution function assigns a probability mass $1/n$ to each observed value of the random vectors $\mathbf{X}_i$ for $i = 1, 2, ..., n$ [9].

The empirical distribution function is the maximum likelihood estimator of the distribution for the observations when no parametric assumptions are made. The bootstrap distribution for $\hat{\theta} - \theta$ is the distribution obtained by generating $\hat{\theta}$ values by sampling independently with replacement from the empirical distribution F_n. The bootstrap estimate of the standard error of $\hat{\theta}$ is then the standard deviation of the bootstrap distribution for $\hat{\theta} - \theta$ [9].

The bootstrap is often categorised as a computer-intensive method. The reason is that in most practical problems, a large number of samples is required where it is deemed to be useful the estimation is complex. In the case of confidence interval and hypothesis testing at least 1000 bootstrap replications are required [9].

The cause of differs between a typical bootstrap sample and the original sample are the random collections of observations. Some observations

may be repeated multiple times and some observations will not appear at all in bootstrap samples [9].

Note that the bootstrap method can fail in several situations such as: the samples size is too small, distribution with infinite moments, estimating extreme values and long-range dependence [9].

3.3 Bootstrap mean estimates

For population distributions with finite first moments, the mean is a natural measure of central tendency. The sample mean is the "best" estimate, and bootstrapping adds nothing to the parametric approach. For some distributions for which the sample mean does not converge in probability to the population mean, we estimate the median instead of the mean since the population median always exists and is consistently estimated by the sample median. Again, the bootstrap adds nothing to the point estimation, but it is useful in estimating standard errors and percentiles [9].

Estimating the bootstrap mean is straightforward, there are only two steps:

1. Take **B** bootstrap samples.

2. Calculate the mean for each bootstrap sample. So we get **B** bootstrap sample means.

3.4 Bootstrap Confidence Intervals

There are three standard methods of construction of confidence intervals using bootstrap for multivariate data [11].

1. Bootstrap Percentile Method:

Suppose we have 1000 bootstrap replications of $\hat{\theta}$, denoted by $(\theta^*_1, \theta^*_2, ..., \theta^*_{1000})$. After ranking them from bottom to top, the bootstrap percentile confidence interval at 95% level of confidence would be $[\theta^*_{(25)}, \theta^*_{(975)}]$. The Boot-

strap Percentile Method is valid when the sampling distribution of $\hat{\theta}$ is approximately symmetric around θ.

2. Centred Bootstrap Percentile Method:

The centred bootstrap percentile confidence interval at coverage level 95% is $[2\hat{\theta} - \theta^*_{(975)}, 2\hat{\theta} - \theta^*_{(25)}]$.

3. Bootstrap-t Method:

The 95% bootstrap-t confidence interval is $[\hat{\theta} - \hat{se} \cdot \hat{t}^{(1-0.05)}, \hat{\theta} - \hat{se} \cdot \hat{t}^{(0.05)}]$.

Where $\hat{se}$ is the estimated standard error of the coefficient in the original model, $\hat{t}^{(1-0.05)}$ denotes the $1 - 0.05$ percentile of the bootstrapped Student's t-test $\hat{t} = (\hat{\theta}^* - \hat{\theta})/\hat{se}_{\hat{\theta}^*}$.

3.5 Bootstrap method for the confidence ellipse of bivariate data

In the first section, we tested the normality of the bivariate data set, and the result is that the data are not normally distributed. Since the data is not normally distributed, it is better to use the bootstrap method to construct the confidence ellipse for the means of the bivariate data rather than using the maximum log-likelihood method to compute the means and variance-covariance matrix and construct a confidence region based on the normality assumption.

The steps to construct the 95% confidence ellipse for the bivariate data are:

1. Take 1000 bootstrap samples, i.e. $\mathbf{B} = 1000$.

2. Calculate the mean for each bootstrap sample.

3. Construct a confidence ellipse using the inequality (2.1).

We use these steps to construct a 95% confidence ellipse for the bootstrap means. There are two plots below, the data we used for the first plot

is the data after deleting outliers using the Euclidean method and the data we used for the second plot is the data after deleting outliers using the Mahalanobis method.

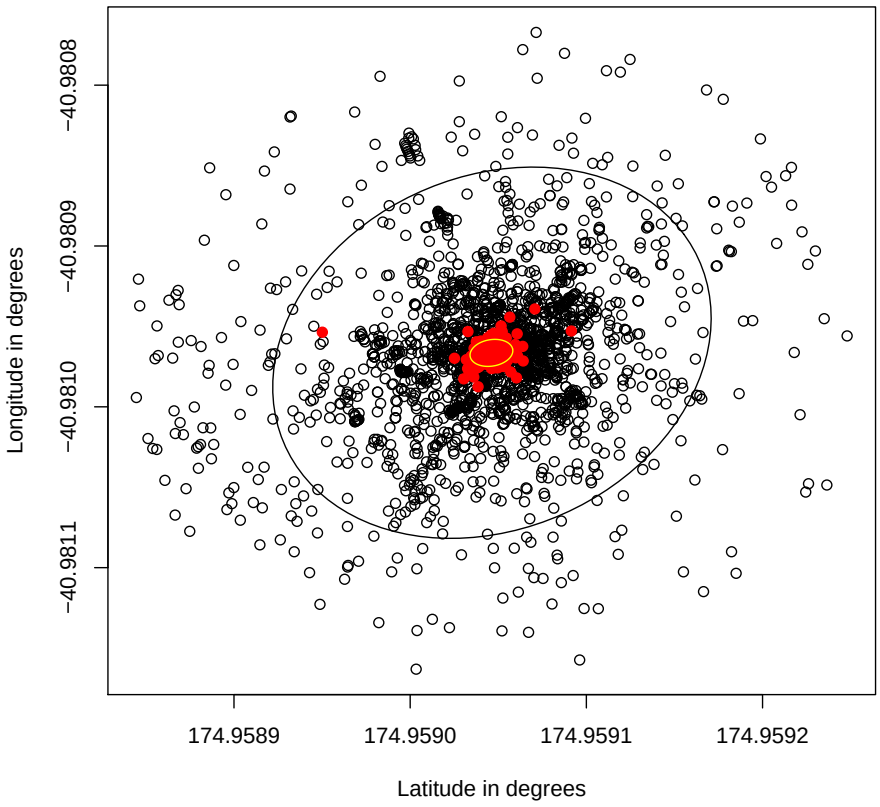


Figure 3.1: Combined bootstrap confidence ellipse and data confidence ellipse after using the Euclidean distance method for deleting outliers

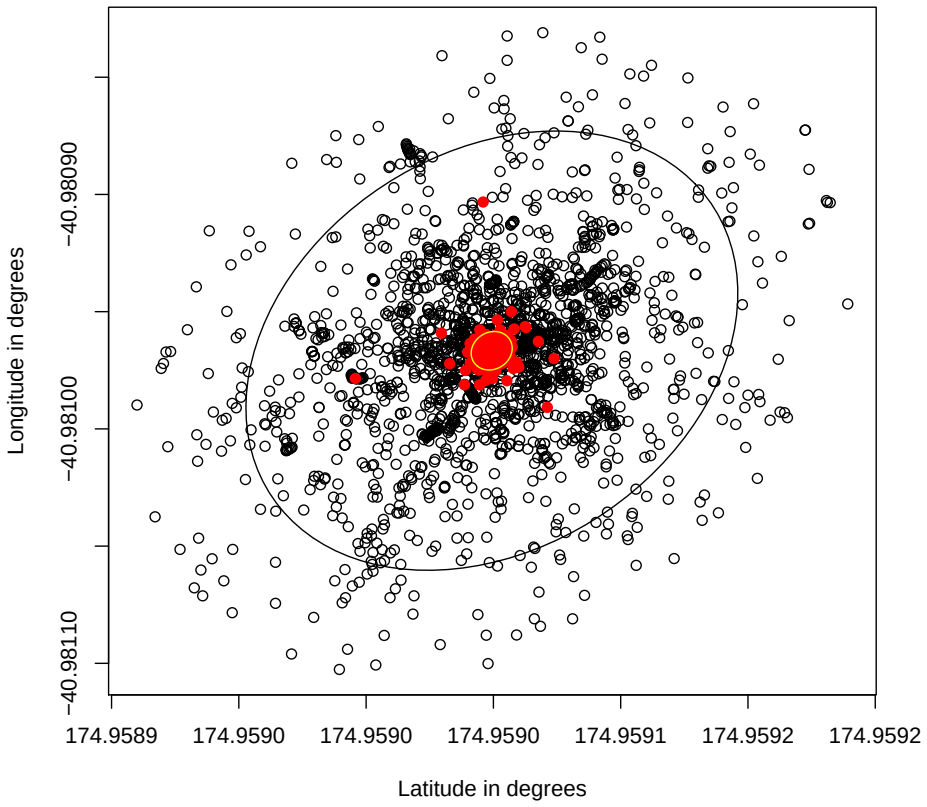


Figure 3.2: Combined bootstrap confidence ellipse and data confidence ellipse after using the Mahalanobis distance method for deleting outliers

The black dots and black circle are from the original data and confidence ellipse, the red dots are the bootstrap samples means, and the yellow circle is the confidence ellipse for the bootstrap sample means. From the plots, we can see the bootstrap confidence is much smaller than the confidence ellipse for the original data means because the bootstrap confidence ellipse uses the variance-covariance matrix of bootstrap samples means instead of the data.

From the R output, when the Euclidean distance method is used, the mean and variance-covariance matrix for the bootstrap sample means are given by:

$$\begin{pmatrix} \bar{X} \\ \bar{Y} \end{pmatrix} = \begin{pmatrix} -40.98097° \\ 174.959° \end{pmatrix}$$

and

$$S = \begin{pmatrix} 2.459720 \times 10^{-11} & 2.757104 \times 10^{-12} \\ 2.757104 \times 10^{-12} & 1.177883 \times 10^{-11} \end{pmatrix}.$$

When the Mahalanobis distance method is used, the mean and variance-covariance matrix for the bootstrap sample means are given by:

$$\begin{pmatrix} \bar{X} \\ \bar{Y} \end{pmatrix} = \begin{pmatrix} -40.98097° \\ 174.959° \end{pmatrix}$$

and

$$S = \begin{pmatrix} 1.082630 \times 10^{-11} & 1.418083 \times 10^{12} \\ 1.418083 \times 10^{12} & 1.123571 \times 10^{11} \end{pmatrix}.$$

Comparing the mean for the bootstrap sample and the mean for the observations, we notice that the two means are the same. The variance-covariance matrix for the bootstrap sample are much smaller than for the observations.

3.6 Bootstrap SCI

There are two methods to construct the simultaneous confidence intervals (SCI) based on the percentile bootstrap approach. The methods are introduced by Mandel and Betensky in 2008 [31].

Suppose that the data $\mathbf{X}$ was generated by a law F and we are interested in SCI for $(\theta_1, ..., \theta_m) = (\theta_1(F), ..., \theta_m(F))$. We want to construct SCI

for $(\theta_1, ..., \theta_m)$ with a simultaneous coverage of $1-\alpha$ based on B bootstrap samples. The algorithms are below [31]:

Algorithm 1

1. Generate B bootstrap samples from $\hat{F}$, an estimate of F. For each sample, $\mathbf{X}_b$, calculate the estimates $\tilde{\theta}_b = (\tilde{\theta}_{b1}, ..., \tilde{\theta}_{bm})$.

2. For each coordinate j, order the bootstrap estimates and denote them by $\tilde{\theta}_{(1j)} < \tilde{\theta}_{(2j)} < ... < \tilde{\theta}_{(Bj)}$. Define $r(b, j)$ to be the rank of $\tilde{\theta}_{bj}$, i.e., $\tilde{\theta}_{bj} = \tilde{\theta}_{(r(b,j)j)}$.

3. Define the sample-b rank $r(b) = \max_j r(b, j)$ to be the largest rank associated with the b'th bootstrap estimate.

4. Calculate $r_{1-\alpha/2}$, the $1-\alpha/2$ percentile of $r(b)$.

5. Take the upper limits of the SCI to be $\tilde{\theta}_{(r_{1-\alpha/2}1)}, ..., \tilde{\theta}_{(r_{1-\alpha/2}m)}$. Construction of the lower limit is analogous.

Algorithm 2

1. Repeat Steps 1 and 2 of Algorithm 1.
2. Define the relative ranks $r^*(b,j) = |r(b,j) - (B+1)/2|$ and their signs $s^*(b,j) = \text{sign}\{r(b,j) - (B+1)/2\}$. Thus, a high $r^*(b,j)$ means an extreme estimate of θ_j, either small $s^*(b,j) = -1$ or large $s^*(b,j) = 1$.
3. Define $r^*(b) = \max_j r^*(b,j)$ to be the largest relative rank of the b'th bootstrap estimate and let $s^*(b)$ be the associated sign. It is possible that the maximum is obtained at several j's. $r^*(b)$ is well defined in such cases but the corresponding sign may not. If this is the case, choose $s^*(b)$ arbitrarily.
4. Let $r(b) = (B+1)/2 + r^*(b)s^*(b)$ be the original rank corresponding to the most discrepant coordinate of the b'th sample.
5. For all j and all b, replace $\tilde{\theta}_{bj}$ with $\tilde{\theta}_{(r(b)j)}$. This yields one rank for each bootstrap estimate with a possibility of ties,i.e., the new estimates are comparable with respect to the relation $>$.
6. Apply Algorithm 1 on the new values generated in Step 5.

In our data, we have X and Y as our bivariate data set that after deleting outliers using the Euclidean method. Since the results from the two deleting methods are similar, we choose only one method here. We choose B to be 1000. Here $j \in \{1,2\}$ and $m = 2$. $\tilde{\theta}$ is the mean for each bootstrap sample. We use Algorithm 1 and Algorithm 2 to find the upper limits and use min in Step 3 in both algorithms to find the lower limits, then use these two limits as diagonal points to construct a rectangle which is the confidence region. We use α is 0.05, so that we construct a 95% confidence region. We use a similar way to do the SCI for our bivariate data, we do

not need to generate bootstrap samples in Algorithm 1 step 1, we use the bivariate data set instead, then order the data and follow the rest of the steps.

The plot for the bivariate data SCI and bootstrap SCI obtained using algorithm 1 is shown below. The black dots are our bivariate data and the red dots are the simulated means using the bootstrap method. The red rectangle is the bivariate data SCI and the blue rectangle is the bootstrap SCI:

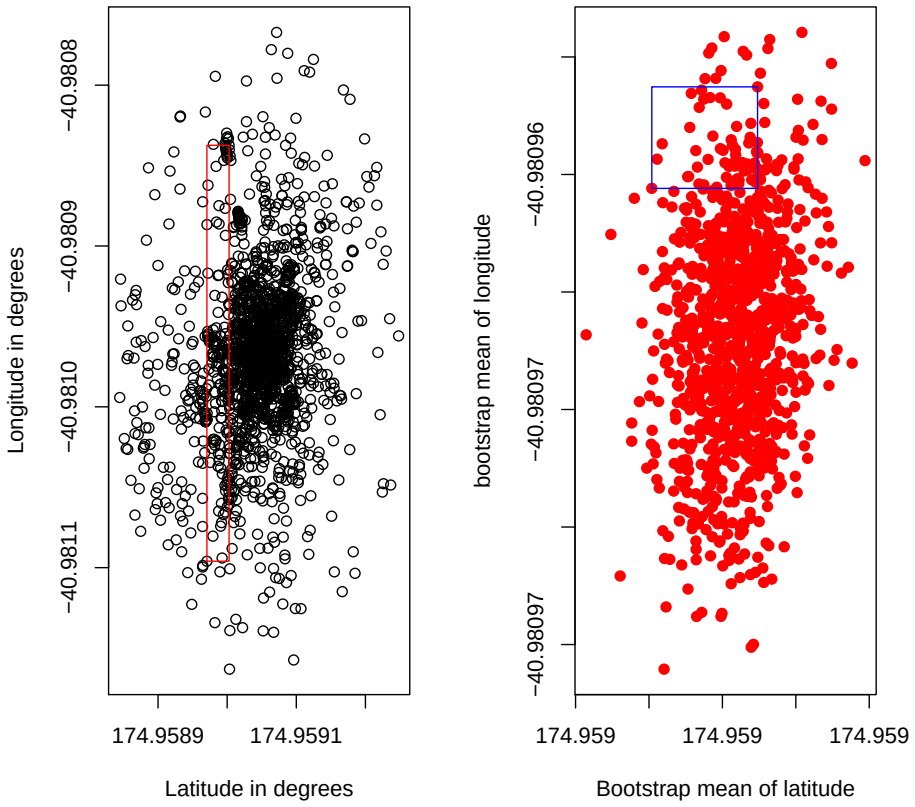


Figure 3.3: Original data confidence region and Bootstrap confidence region using algorithm 1

The plot for the bivariate data SCI and bootstrap SCI obtained using algorithm 2 is shown below. The red rectangle is the bivariate data SCI and the blue rectangle is the bootstrap SCI:

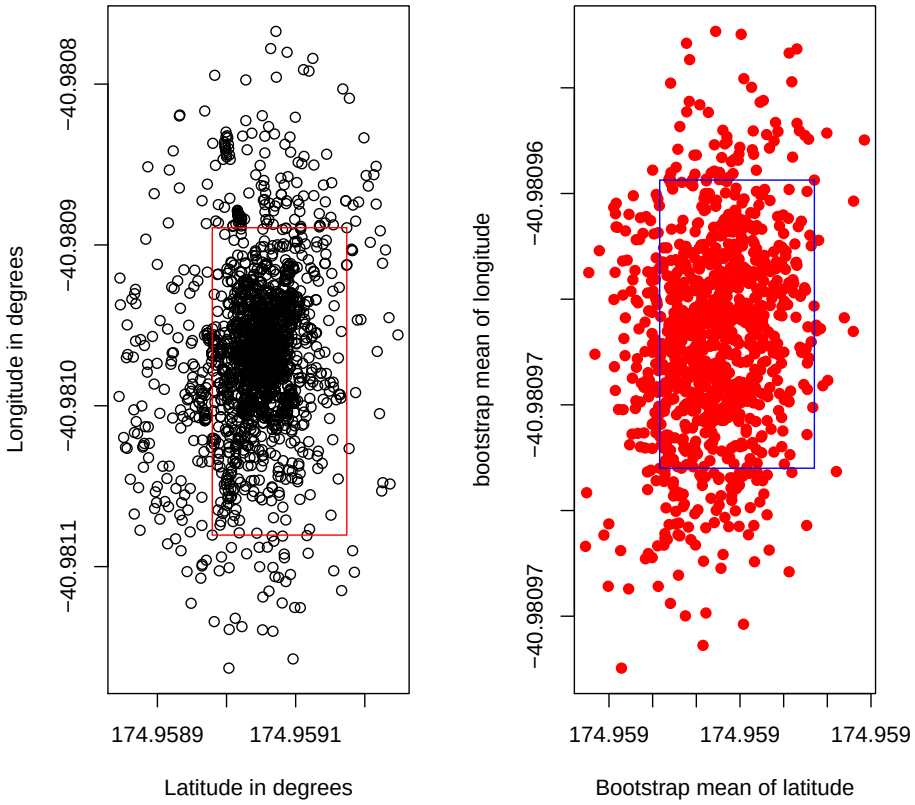


Figure 3.4: Original data confidence region and Bootstrap confidence region using algorithm 2

The algorithm based on the simple percentile bootstrap method does not always work well. The difference between the two algorithms is Algorithm 2 uses relative ranks but Algorithm 1 uses the ranks themselves [31].

Algorithms 1 and 2 implicitly assume that there are no ties. In the process for our bivariate data and bootstrap sample mean data, there are ties. The two algorithms for constructing simultaneous confidence intervals do not seem as good as constructing the confidence ellipse, this is because the confidence ellipse is using the bivariate set together and the covariance for the bivariate data set, but the SCI calculates the upper and lower limits separately for each variable and we have some ties in our data.

3.7 Summary

In this chapter, the bootstrap method is used because we found that the data is not normally distributed using the Shapiro-Wilk's statistic. We use the bootstrap method to estimate the means of the bivariate data and construct the confidence ellipse by using bootstrap sample means and bootstrap sample covariance matrix. We also construct the confidence rectangle for the bivariate data, based on the percentile bootstrap approach.

Chapter 4

Multivariate Linear Regression Models

In this chapter, the aim is to determine whether position measurements and accuracy of the GPS readings are related to other characteristics such as the number of satellites, navigation type and others. We use multivariate linear regression models to determine this. We start by plotting latitude and longitude without outliers removed against the other characteristics, with the aim of determining whether outliers are related to any other characteristics. If outliers systematically occur at specific values of a given characteristic, then that can be used to inform the process of outlier removal. The characteristics include GPSTOW, Nav Type, Num Sats, and Altitude.MSL (see section 2.1 for definition). The 3D plots are below:

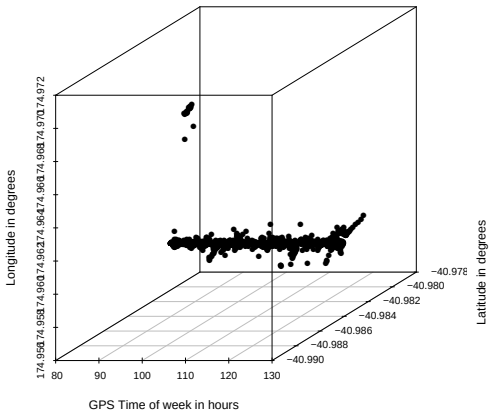


Figure 4.1: Latitude, Longitude against GPSTOW

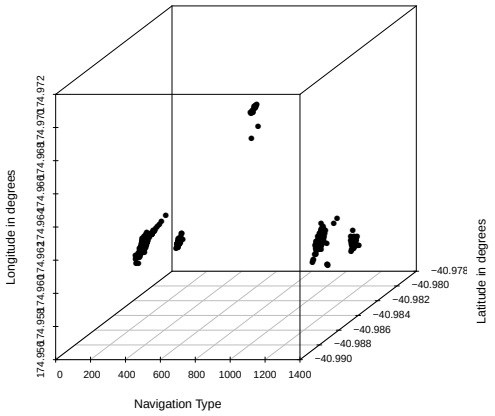


Figure 4.2: Latitude, Longitude against Nav Type

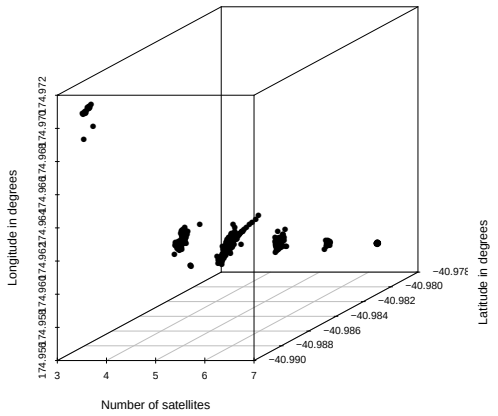


Figure 4.3: Latitude, Longitude against Num Sats

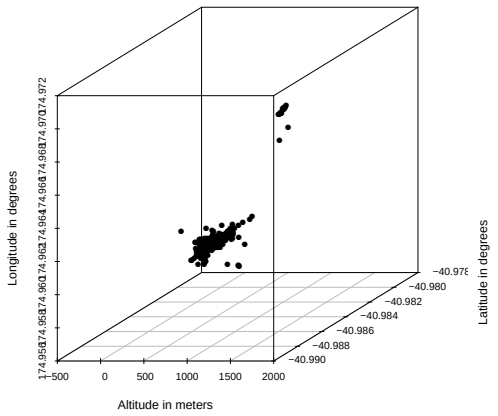


Figure 4.4: Latitude, Longitude against Altitude.MSL

From these plots, we can see that GPSTOW has a linear relationship with Latitude and Longitude. We need to investigate more by fitting data in multivariate linear regression model.

The 3D plots below are replotted after using the Euclidean distance method for removing outliers.

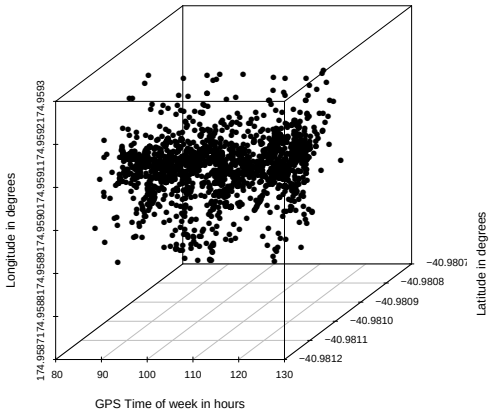


Figure 4.5: Latitude, Longitude against GPSTOW

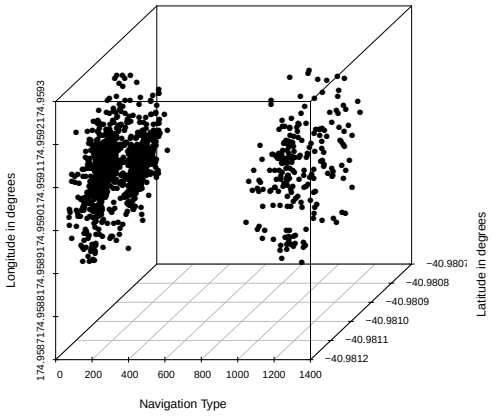


Figure 4.6: Latitude, Longitude against Nav Type

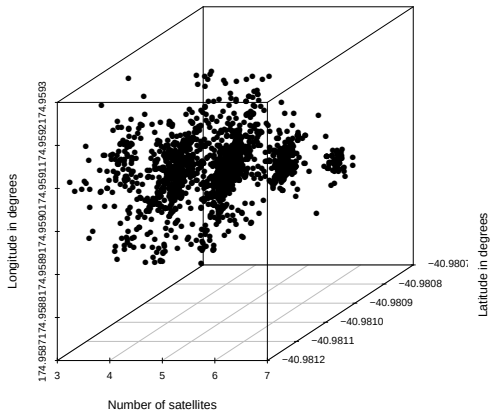


Figure 4.7: Latitude, Longitude against Num Sats

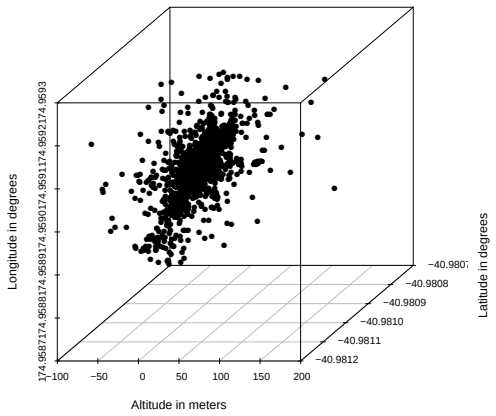


Figure 4.8: Latitude, Longitude against Altitude.MSL

From these plots, we can see that the extreme outliers are occur under the following condition: Number of satellites is 3, Navigation type between 106 and 1016, Altitude is above 200 meters.

4.1 The Basic Principle

The linear regression model with a single response variable is:

$$Y = \beta_0 + \beta_1 z_1 + \cdots + \beta_r z_r + \varepsilon$$

where Y is the response variable, $z_1 \cdots z_r$ are the predictor variables, $\beta_0 \cdots \beta_r$ are the unknown regression coefficients and ε is the error term.

If we have m response variables and a single set of r predictor variables $z_1 \cdots z_r$, then it is called the multivariate multiple linear regression model. Each of the m responses is assumed to follow its own regression model, so that

$$\begin{aligned} Y_1 &= \beta_{01} + \beta_{11} z_1 + \cdots + \beta_{r1} z_r + \varepsilon_1 \\ Y_2 &= \beta_{02} + \beta_{12} z_1 + \cdots + \beta_{r2} z_r + \varepsilon_2 \\ &\vdots \\ Y_m &= \beta_{0m} + \beta_{1m} z_1 + \cdots + \beta_{rm} z_r + \varepsilon_m \end{aligned}$$

The error term $\varepsilon' = [\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_m]$ has $E(\varepsilon) = 0$ and $\text{Var}(\varepsilon) = \Sigma$. Thus, the error terms associated with different response variables may be correlated.

Conceptually, we can let $[z_{j0}, z_{j1}, \cdots, z_{jr}]$ denote the values of the predictor variables for the j^{th} observation, $j = 1, ..n$ and $\mathbf{Y}_j' = [Y_{j1}, Y_{j2}, \cdots, Y_{jm}]$ be the response variables, and let $\boldsymbol{\varepsilon}_j' = [\varepsilon_{j1}, \varepsilon_{j2}, \cdots, \varepsilon_{jm}]$ be the errors. Thus, we have an $n \times (r+1)$ design matrix

$$\mathbf{Z} = \begin{bmatrix} z_{10} & z_{11} & \cdots & z_{1r} \\ z_{20} & z_{21} & \cdots & z_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ z_{n0} & z_{n1} & \cdots & z_{nr} \end{bmatrix}.$$

If we now set

$$\mathbf{Y} = \begin{bmatrix} Y_{11} & Y_{12} & \cdots & Y_{1m} \\ Y_{21} & Y_{22} & \cdots & Y_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ Y_{n1} & Y_{n2} & \cdots & Y_{nm} \end{bmatrix} = [\mathbf{Y}_{(1)} | \mathbf{Y}_{(2)} | \cdots | \mathbf{Y}_{(m)}]$$

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_{01} & \beta_{02} & \cdots & \beta_{0m} \\ \beta_{11} & \beta_{12} & \cdots & \beta_{1m} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{r1} & \beta_{r2} & \cdots & \beta_{rm} \end{bmatrix} = [\boldsymbol{\beta}_{(1)}|\boldsymbol{\beta}_{(2)}|\cdots|\boldsymbol{\beta}_{(m)}],$$

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_{11} & \varepsilon_{12} & \cdots & \varepsilon_{1m} \\ \varepsilon_{21} & \varepsilon_{22} & \cdots & \varepsilon_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \varepsilon_{n1} & \varepsilon_{n2} & \cdots & \varepsilon_{nm} \end{bmatrix} = [\boldsymbol{\varepsilon}_{(1)}|\boldsymbol{\varepsilon}_{(2)}|\cdots|\boldsymbol{\varepsilon}_{(m)}] = \begin{bmatrix} \boldsymbol{\varepsilon}'_{(1)} \\ - \\ \boldsymbol{\varepsilon}'_{(2)} \\ - \\ \vdots \\ - \\ \boldsymbol{\varepsilon}'_{(m)} \end{bmatrix}',$$

the multivariate linear regression model is

$$\boldsymbol{Y} = \boldsymbol{Z}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

with $E(\boldsymbol{\varepsilon}_{(i)}) = \mathbf{0}$ and $\mathrm{Cov}(\boldsymbol{\varepsilon}_{(i)}\boldsymbol{\varepsilon}_{(k)}) = \sigma_{ik}\mathbf{I}$, for $i, k = 1, 2, ..., m$.

The m observations on the jth trial have covariance matrix $\Sigma = \{\sigma_{ik}\}$, but observations from different trials are uncorrelated. Here $\boldsymbol{\beta}$ and σ_{ik} are unknown parameters; the design matrix $\mathbf{Z}$ has jth row $[z_{j0}, z_{j1}, ..., z_{jr}]$.

Given the outcomes $\mathbf{Y}$ and the values of the predictor variables $\mathbf{Z}$ with full column rank, we determine the least squares estimates $\hat{\boldsymbol{\beta}}_{(i)}$ exclusively from the observations $\mathbf{Y}_{(i)}$ on the ith response. In conformity with the single-response solution, we take

$$\hat{\boldsymbol{\beta}}_{(i)} = (\mathbf{Z}'\mathbf{Z})^{-1}\mathbf{Z}'\mathbf{Y}_{(i)}$$

Collecting these univariate least squares estimates, we obtain

$$\hat{\boldsymbol{\beta}} = (\mathbf{Z}'\mathbf{Z})^{-1}\mathbf{Z}'\mathbf{Y}$$

For any choice of parameters $\mathbf{B} = [\mathbf{b}_{(1)}|\mathbf{b}_{(2)}|\cdots|\mathbf{b}_{(m)}]$, the matrix of errors is $\mathbf{Y} - \mathbf{Z}\mathbf{B}$.

The error sum of squares and crossproducts matrix is:

$$(\mathbf{Y}-\mathbf{ZB})'(\mathbf{Y}-\mathbf{ZB}) =$$
$$\begin{bmatrix} (\mathbf{Y}_{(1)}-\mathbf{Zb}_{(1)})'(\mathbf{Y}_{(1)}-\mathbf{Zb}_{(1)}) & \cdots & (\mathbf{Y}_{(1)}-\mathbf{Zb}_{(1)})'(\mathbf{Y}_{(m)}-\mathbf{Zb}_{(m)}) \\ \vdots & & \vdots \\ (\mathbf{Y}_{(m)}-\mathbf{Zb}_{(m)})'(\mathbf{Y}_{(1)}-\mathbf{Zb}_{(1)}) & \cdots & (\mathbf{Y}_{(m)}-\mathbf{Zb}_{(m)})'(\mathbf{Y}_{(m)}-\mathbf{Zb}_{(m)}) \end{bmatrix}$$

The section $\mathbf{b}_{(i)} = \hat{\boldsymbol{\beta}}_{(i)}$ minimizes the ith diagonal sum of squares $(\mathbf{Y}_{(i)} - \mathbf{Zb}_{(i)})'(\mathbf{Y}_i - \mathbf{Zb}_{(i)})$. Consequently, tr$[(\mathbf{Y}-\mathbf{ZB})'(\mathbf{Y}-\mathbf{ZB})]$ is minimized by the choice $\mathbf{B} = \hat{\boldsymbol{\beta}}$. Also, the generalized variance $|(\mathbf{Y}-\mathbf{ZB})'(\mathbf{Y}-\mathbf{ZB})|$ is minimized by the least squares estimates $\hat{\boldsymbol{\beta}}$.

So we have matrix of predicted values

$$\hat{\mathbf{Y}} = \mathbf{Z}\hat{\boldsymbol{\beta}} = \mathbf{Z}(\mathbf{Z}'\mathbf{Z})^{-1}\mathbf{Z}'\mathbf{Y}$$

and we have a resulting matrix of residuals

$$\hat{\boldsymbol{\varepsilon}} = \mathbf{Y} - \hat{\mathbf{Y}} = [\mathbf{I} - \mathbf{Z}(\mathbf{Z}'\mathbf{Z})^{-1}\mathbf{Z}']\mathbf{Y}$$

Note that the orthogonality conditions among residuals, predicted values, and columns of the design matrix which hold in the univariate case are also true in the multivariate case because

$$\mathbf{Z}'[\mathbf{I} - \mathbf{Z}(\mathbf{Z}'\mathbf{Z})^{-1}\mathbf{Z}'] = \mathbf{Z}' - \mathbf{Z}' = \mathbf{0}$$

which means the residuals are perpendicular to the columns of the design matrix

$$\mathbf{Z}'\hat{\boldsymbol{\varepsilon}} = \mathbf{Z}'[\mathbf{I} - \mathbf{Z}(\mathbf{Z}'\mathbf{Z})^{-1}\mathbf{Z}']\mathbf{Y} = \mathbf{Z}' - \mathbf{Z}' = \mathbf{0}$$

and to the predicted values

$$\hat{\mathbf{Y}}'\hat{\boldsymbol{\varepsilon}} = \hat{\boldsymbol{\beta}}'\mathbf{Z}'[\mathbf{I} - \mathbf{Z}(\mathbf{Z}'\mathbf{Z})^{-1}\mathbf{Z}']\mathbf{Y} = 0.$$

Furthermore, because

$$\mathbf{Y} = \hat{\mathbf{Y}} + \hat{\boldsymbol{\varepsilon}},$$

we have

$$\mathbf{Y}'\mathbf{Y} = \hat{\mathbf{Y}}'\hat{\mathbf{Y}} + \hat{\boldsymbol{\varepsilon}}'\boldsymbol{\varepsilon}.$$

4.2 Inference in Multivariate Regression

The least squares estimators

$$\hat{\boldsymbol{\beta}} = [\hat{\boldsymbol{\beta}}_{(1)}|\hat{\boldsymbol{\beta}}_{(2)}|\cdots|\hat{\boldsymbol{\beta}}_{(m)}]$$

of the multivariate regression model have the following properties:

- $E[\hat{\boldsymbol{\beta}}_{(i)}] = \boldsymbol{\beta}_{(i)}$ or $E[\hat{\boldsymbol{\beta}}] = \boldsymbol{\beta}$.
- $\text{Cov}(\hat{\boldsymbol{\beta}}_{(i)}, \hat{\boldsymbol{\beta}}_{(k)}) = \sigma_{ik}(\mathbf{Z}'\mathbf{Z})^{-1}$, $i, k = 1, ..., m$.
- $E(\hat{\boldsymbol{\varepsilon}}) = 0$ and $E(\frac{1}{n-r-1}\hat{\boldsymbol{\varepsilon}}'\hat{\boldsymbol{\varepsilon}}) = \boldsymbol{\Sigma}$.

Also $\hat{\boldsymbol{\varepsilon}}$ and $\hat{\boldsymbol{\beta}}$ are uncorrelated.

This means that, for any observation $\mathbf{z}_0$

$$\mathbf{z}_0'\hat{\boldsymbol{\beta}} = [\mathbf{z}_0'\hat{\boldsymbol{\beta}}_{(1)}|\mathbf{z}_0'\hat{\boldsymbol{\beta}}_{(2)}|\cdots|\mathbf{z}_0'\hat{\boldsymbol{\beta}}_{(m)}]$$

is an unbiased estimator, i.e.,

$$\mathrm{E}[\mathbf{z}_0'\hat{\boldsymbol{\beta}}] = \mathbf{z}_0'\boldsymbol{\beta}$$

We can also determine from these properties that the estimation errors

$$\mathbf{z}_0'\boldsymbol{\beta}_{(i)} - \mathbf{z}_0'\hat{\boldsymbol{\beta}}_{(i)}$$

have covariances

$$E[\mathbf{z}_0'(\boldsymbol{\beta}_{(i)} - \hat{\boldsymbol{\beta}}_{(i)})(\boldsymbol{\beta}_{(k)} - \hat{\boldsymbol{\beta}}_{(k)})'\mathbf{z}_0] = \boldsymbol{\sigma}_{ik}\mathbf{z}_0'(\mathbf{Z}'\mathbf{Z})^{-1}\mathbf{z}_0.$$

For a multivariate regression model with full rank $(\mathbf{Z}) = r + 1, n \geq r + 1 + m$, and normally distributed error $\boldsymbol{\varepsilon}$,

$$\hat{\boldsymbol{\beta}} = (\mathbf{Z}'\mathbf{Z})^{-1}\mathbf{Z}'\mathbf{Y}$$

is the maximum likelihood estimator of $\boldsymbol{\beta}$ and

$$\hat{\boldsymbol{\beta}} \sim N(\boldsymbol{\beta}, \boldsymbol{\Sigma})$$

where $\mathbf{\Sigma} = Cov(\hat{\boldsymbol{\beta}}_{(i)}, \hat{\boldsymbol{\beta}}_{(k)}) = \sigma_{ik}(\mathbf{Z}'\mathbf{Z})^{-1}, i, k = 1, ..., m.$

Also, the maximum likelihood estimator of $\hat{\boldsymbol{\beta}}$ is independent of the maximum likelihood estimator of the positive definite matrix $\mathbf{\Sigma}$ given by

$$\mathbf{\Sigma} = \frac{1}{n}\hat{\boldsymbol{\varepsilon}}'\hat{\boldsymbol{\varepsilon}} = \frac{1}{n}(\mathbf{Y} - \mathbf{z}\hat{\boldsymbol{\beta}})'(\mathbf{Y} - \mathbf{z}\hat{\boldsymbol{\beta}})$$

and

$$n\hat{\mathbf{\Sigma}} \sim W_{p,n-r-1}(\mathbf{\Sigma})$$

all of which provide additional support for using the least squares estimate when the errors are normally distributed $\hat{\boldsymbol{\beta}}$ and $n^{-1}\hat{\boldsymbol{\varepsilon}}'\hat{\boldsymbol{\varepsilon}}$ are the maximum likelihood estimators of $\boldsymbol{\beta}$ and $\mathbf{\Sigma}$. $W_{p,n-r-1}(\mathbf{\Sigma})$ is a wishart distribution with $n - r - 1$ degree of freedom.

These results can be used to develop likelihood ratio tests for the multivariate regression parameters.

4.3 Model selection

We use model selection to determine which, if any characteristics are related to longitude and latitude.

Statistical modelling is a critical tool in scientific research. Statistical models are used to understand phenomena with uncertainty, to determine the structure of complex systems, to control such systems and to make reliable predictions in various natural and social science fields [26].

There are many ways to measure the goodness of fit of a statistical model and to find the best fitted model from a set of models. Akaike information criterion (AIC) and Bayesian information criterion (BIC) are two widely used model selection criteria. We start by investigating the performance of AIC and BIC for the linear regression models and then apply it to the multivariate linear regression case in our data.

4.3.1 AIC

The Kullback-Leibler (K-L) information is a function denoted as "I" for information. This function has two arguments: f represents the full reality and g is a model. Then, the K-L information $I(f, g)$ is the " information" lost when the model g is used to approximate the full reality, f. We need to find a candidate model that minimizes $I(f, g)$, over the hypothesis test, represented by models. The model with the smallest information loss would be the best model and therefore would represent the best hypothesis [2].
The K-L information is defined by integral for continuous distributions (eg., the normal or gamma) [2]:

$$I(f, g) = \int f(X) \log \Big(\frac{f(X)}{g(X|\theta)} \Big) dX.$$

The K-L information is defined by summation for discrete distributions (eg., Poisson, binomial, or multinomial) [2]:

$$I(f, g) = \sum_{i=1}^{k} p_i \log \Big(\frac{p_i}{\pi_i} \Big).$$

Here, there are k possible outcomes of the underlying random variables; the true probability of the ith outcome is given by p_i, while the $\pi_1, ..., \pi_k$ constitutes the approximate probability distribution.

Akaike's main steps started by using a property of logarithms to rewrite K-L information for continuous distribution as [2]

$$I(f, g) = \int f(X) \log(f(X)) dX - \int f(X) \log(g(X|\theta)) dX.$$

Thus, K-L information can be expressed as [2]

$$I(f, g) = E_f[\log(f(X))] - E_f[\log(g(X|\theta))],$$

each expectation with respect to the true distribution f.
The expectation of $[\log(f(X))]$ does not change from model to model; it is a constant. Thus, we are only left with the second expectation,

$$I(f, g) - C = -E_f[\log(g(X|\theta))].$$

Akaike's discovery of a relationship between the K-L information and the maximized log-likelihood has allowed major practical and theoretical advances in model selection and the analysis of complex data sets [2].
Akaike found that he could not estimate the K-L, but he could estimate the expectation of the K-L information. This second expectation is over the data (denote these data as y)

$$E_f[\log(g(X|\hat{\theta}))],$$

where the estimates $\hat{\theta}$ are based on the data (y).
He showed that the critical issue became the estimation of

$$E_y E_X[\log(g(X|\hat{\theta}(y))].$$

This double expectation, both with respect to truth f, is the target of all model selection approaches based on K-L information [2].
Akaike realized that this complex entity was closely related to the log-likelihood function at its maximum. However, the maximized log-likelihood is biased upward as an estimator of this quantity. He found that under certain conditions, this bias is approximately equal to K, the number of estimable parameters in the approximating model. Thus under mild conditions, an asymptotically unbiased estimator of [2]

$$E_y E_X[\log(g(X|\hat{\theta}(y))] \quad \text{is} \quad \log(\mathcal{L}(\hat{\theta}|\text{data})) - K.$$

Akaike's final step defined "an information criterion" (AIC) by multiplying both terms by -2. Thus, both terms in $\log(\mathcal{L}(\hat{\theta}|\text{data}) - K$ were multiplied by -2 to get [2]

$$AIC = -2\log(\mathcal{L}(\hat{\theta})|data) + 2K.$$

For a univariate linear regression model, the formulae for AIC can be written as:

$$AIC = 2K - 2\ln(L)$$

where $2K$ is a penalty term, K is the number of parameters in the model, and L is the maximized value of the likelihood function for the estimated model.
Given any two estimated models, the model with lower value of AIC is the preferred one.
AIC is an estimator of relative expected Kullback-Leibler information and it is applicable for both nested and non nested models [7]. Another way of writing $\ln(L)$ is:

$$\ln(L) = -\frac{n}{2}(\log(2\pi) + \log(RSS/n) + 1),$$

This equation is used for Gaussian distribution, where RSS is a residual sum of squares:

$$RSS = \sum_{i=1}^{n}(y_i - f(x_i))^2$$

From the formula, we can see if we increase the sample size n , $\ln(L)$ will decrease, then AIC and BIC will increase. If we increase the variance, the RSS will increase and in turn the $\ln(L)$ will decrease while AIC and BIC will increase.

Extension of the AIC criterion to the multivariate linear regression model has been developed [40].

The multivariate regression candidate model is defined by

$$\mathbf{Y} = \mathbf{Z}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

where the rows of the matrix $\mathbf{Y}$ of dimension $n \times m$ correspond to m response variables on each of n individuals, $\mathbf{Z}$ is an $n \times (r+1)$ known matrix of covariance values, and $\boldsymbol{\beta}$ is an $(r+1) \times m$ matrix of unknown regression parameters. The rows of the error matrix $\boldsymbol{\varepsilon}$ of dimension $n \times m$ are assumed to be independent, with identical $N_m(0, \Sigma_m)$ distributions where Σ_m is a $m \times m$ matrix of error covariance.

In this case, the total number of unknown parameters in the model is

$(r+1)m$ for $\boldsymbol{\beta}$ and $m(m+1)/2$ for the covariance matrix Σ_m. Thus, AIC is

$$\mathrm{AIC} = -2\ln(L) + 2\{(r+1)m + m(m+1)/2\}$$

4.3.2 BIC

BIC is also a criterion to measure the goodness of fit of a statistical model. It is an evaluation criterion for models defined in terms of their posterior probability. It is closely related to AIC. The formula for BIC is:

$$\mathrm{BIC} = K\ln(n) - 2\ln(L)$$

$K\ln(n)$ is the penalty term, K is the number of parameters in the model, n is the number of observations or sample size and L is the maximized value of the likelihood function for the estimated model.
Given any two estimated models, the model with a lower value of BIC is the preferred one.
The penalty term in BIC is larger than in AIC when the sample size n is bigger than 7.

For multivariate linear regression models, the BIC is given by:

$$\mathrm{BIC} = -2\ln(L) + \ln(n)\{(r+1)m + m(m+1)/2\}$$

4.4 Results

AIC and BIC are two common information criteria for the model selection. But some problems have been found as well.

For univariate autoregressive (AR) model selection based on small samples, Hurvich and Tsai (1989) have shown that AIC can in fact be quite biased, sometimes leading to severe over fitting, and have presented a corrected version (AIC) which is more nearly unbiased and consequently tends to select much better models than AIC [19].

AIC has been proposed as approximately unbiased estimators for their risks or underlying criterion functions. For selecting multivariate linear regression models, the modified AIC reduces bias in situations where the collection of candidate models includes both underspecified and over specified models. In a simulation study it is verified that the modified AIC provides better approximations to their risk functions, and better model selection than AIC [15].

The model selection literature has been generally poor at reflecting the deep foundations of the Akaike information criterion (AIC) and at making appropriate comparisons to the Bayesian information criterion (BIC). There is a clear philosophy, a sound criterion based in information theory, and a rigorous statistical foundation for AIC. AIC can be justified as Bayesian using a savvy prior on models that is a function of sample size and the number of model parameters. Furthermore, BIC can be derived as a non-Bayesian result. Therefore, arguments about using AIC versus BIC for model selection cannot be from a Bayes versus frequentist perspective [7].

Ichikawa (1998)'s simulation results in a factor analysis indicated that the ability of AIC to select a true model rapidly increased with sample size but at larger sample size it continued to exhibit a slight tendency to select complex models. Similarly, Markon and Krueger (2004) reviewed existing work on factor analysis and noted that AIC performs relatively well in small samples, but is inconsistent and does not improve in performance in large samples whilst BIC in contrast appears to perform relatively poorly in small samples, but is consistent and relatively poor in larger sample sizes [1].

Before applying AIC and BIC to our data, we use simulations to test reliability and stability of these criteria.

4.4.1 Univariate linear regression models

Simulations on the effect of sample size and variance on AIC and BIC for linear regression models:

First, set the true model. The true model is

$$Y_i = \alpha + \beta_1 X_{i1} + \beta_2 X_{i2} + \beta_3 X_{i3} + \varepsilon_i.$$

We choose $\alpha = 0.5$, $\beta_1 = 2, \beta_2 = 10, \beta_3 = 10, X_{i1}, X_{i2}, X_{i3}$ are all normally distributed with $X_{i1} \sim N(5,5)$, $X_{i2} \sim N(10,5)$, $X_{i3} \sim N(20,5)$, and $\varepsilon_i \sim N(0, \sigma^2)$.
Secondly, choose different sample size. The sample sizes are 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100. And also choose different values of σ^2, so that $\sigma^2 \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 2, 5\}$.
Thirdly, simulate X_1, X_2, X_3 and we can get values of $\mathbf{Y}$ by using the model above.
Fourth, fit the model with the true model (i.e. Model 1) and other models using the R function "lm". There are 7 models to be fitted, they are:

Model 1: $Y_i = \alpha + \beta_1 X_{i1} + \beta_2 X_{i2} + \beta_3 X_{i3} + \varepsilon_i$
Model 2: $Y_i = \alpha + \beta_1 X_{i1} + \beta_2 X_{i2} + \varepsilon_i$
Model 3: $Y_i = \alpha + \beta_2 X_{i2} + \beta_3 X_{i3} + \varepsilon_i$
Model 4: $Y_i = \alpha + \beta_1 X_{i1} + \beta_3 X_{i3} + \varepsilon_i$
Model 5: $Y_i = \alpha + \beta_1 X_{i1} + \varepsilon_i$
Model 6: $Y_i = \alpha + \beta_2 X_{i2} + \varepsilon_i$
Model 7: $Y_i = \alpha + \beta_3 X_{i3} + \varepsilon_i$

and get AIC and BIC for each model.

From the outputs, for each sample size and variance, most of the true models have the smallest values for AIC and BIC, which means the true model fits best for the data Y, that is what we expect. Only when the variance is 5 or sample size is 5 or 10,the smallest value for AIC and BIC are not from the true model fit, but the true model is still the best fit model since the difference between the value of AIC for a true model fit and the

model with smallest value for AIC is less than 2.
From the outputs, for true models, when $\sigma^2 = 0.1, 0.2$, increasing sample size implies a decrase in AIC and BIC. When $\sigma^2 \geq 0.3$, the trends for AIC and BIC are increasing.
From the outputs B.4, we can easily see when we have a constant sample size, increasing the value of variance will increase AIC and BIC.

The graphs are put together and the aim is to have a clearer understanding about the effect of sample size and variance. The first graph is to compare AIC and BIC when variance is constant, testing the effects on sample size. The blue dots are AIC and red dots are BIC.

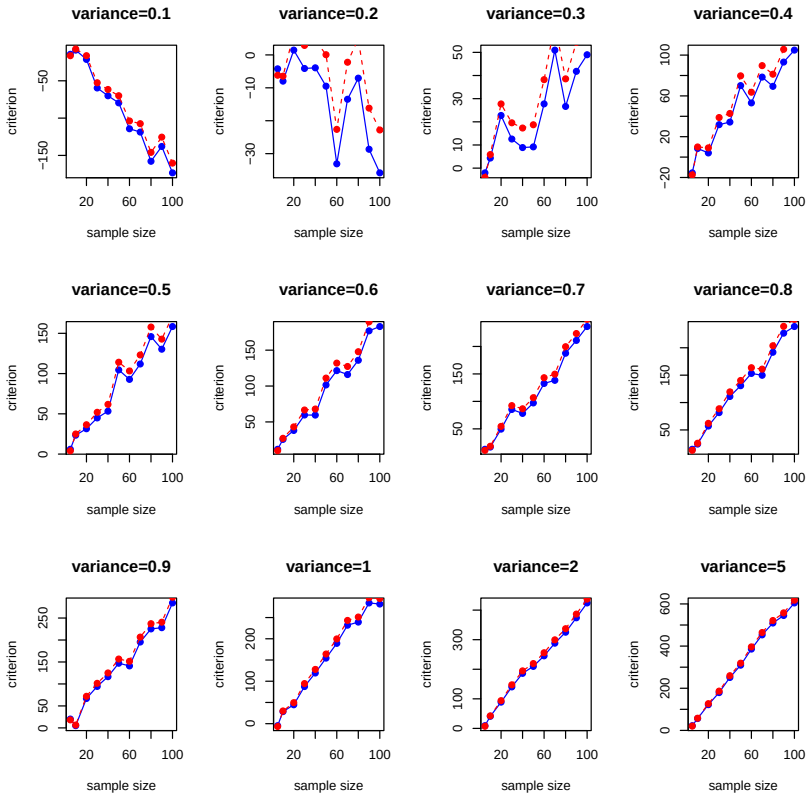


Figure 4.9: The effects on sample size for AIC and BIC

The second graph is to show the effects on variance for AIC and BIC when sample size is constant.

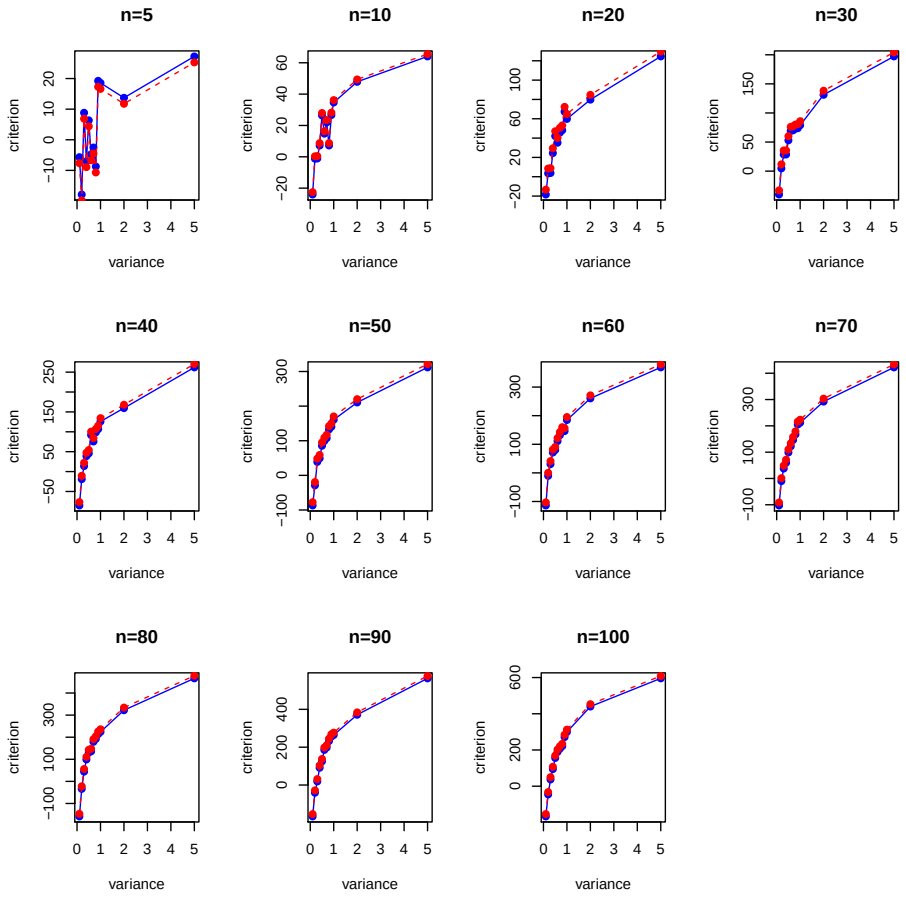


Figure 4.10: The effects on variance for AIC and BIC

The next two graphs are AIC against (variance/sample size) and BIC against (variance/ sample size).

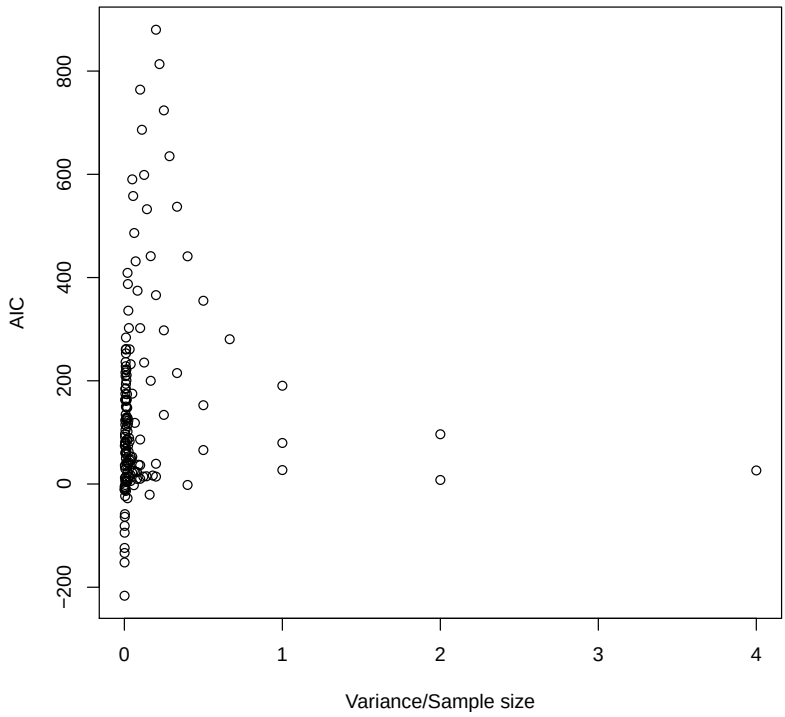


Figure 4.11: AIC against (variance/sample size)

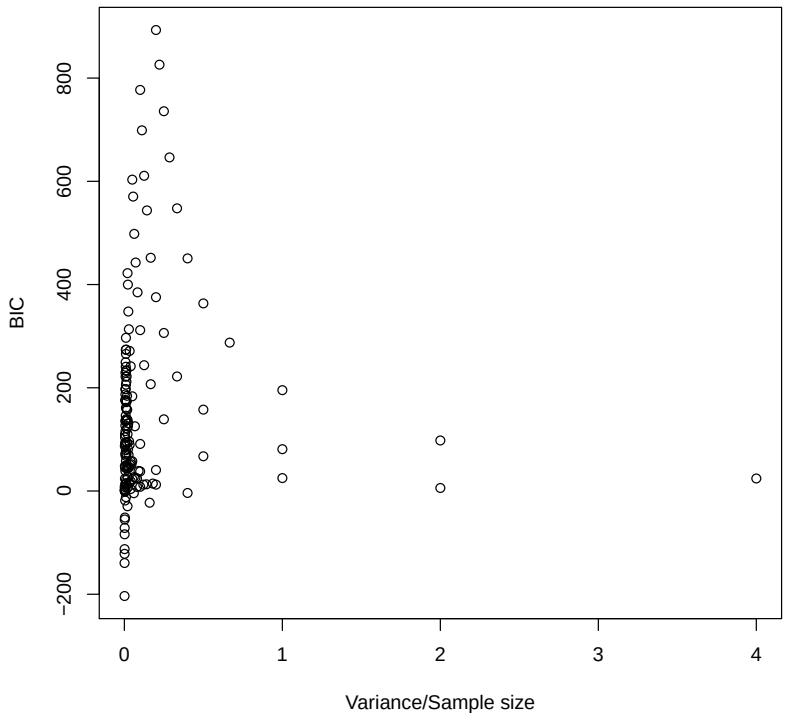


Figure 4.12: BIC against (variance/sample size)

The last two graphs are 3D Scatter plots for AIC and BIC with different sample sizes and variances.

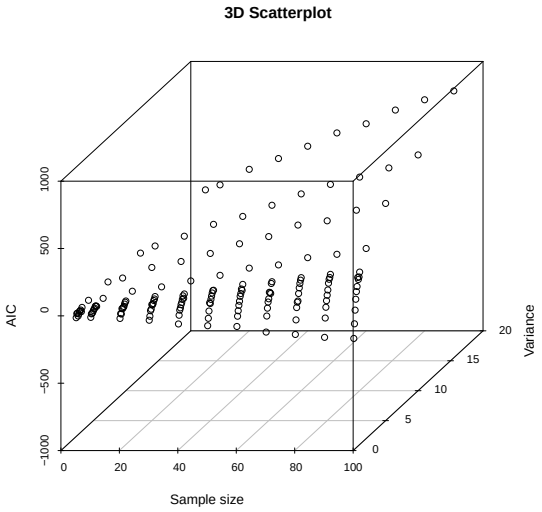


Figure 4.13: AIC with different sample size and variance in 3D

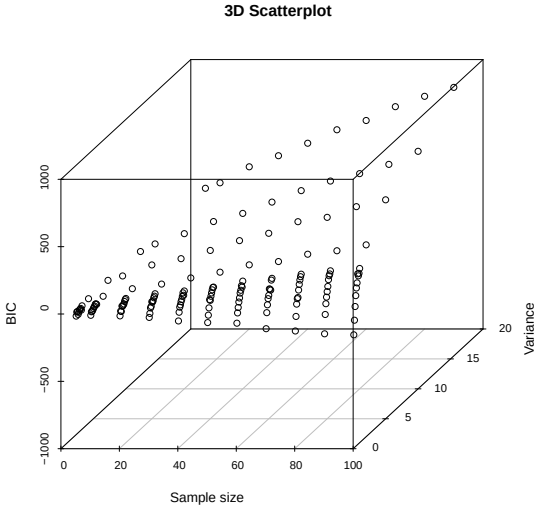


Figure 4.14: BIC with different sample size and variance in 3D

From these graphs, we can see for a small variance, eg. $\sigma^2 = 0.1$, increasing the sample size will give smaller AIC and BIC, but when $\sigma^2 \geq 0.3$, increasing the sample size will give larger AIC and BIC. AIC and BIC are not consistent with variance when changing the sample size. From Figure 4.13 and Figure 4.14, we can see when the values of σ^2/n are bigger, the values of AIC and BIC are close to 0. When the values of σ^2/n close to 0, the values of AIC and BIC are spread between 1000 and -200.

Three typical tables are given below for comparing the proportion of picking the true models with different sample sizes and variances,all remaining tables are inserted in appendix B.4.

	methods	proportion
$n = 5, \sigma^2 = 20$	AIC	0.619
	BIC	0.649
$n = 10, \sigma^2 = 20$	AIC	0.577
	BIC	0.531
$n = 50, \sigma^2 = 20$	AIC	0.974
	BIC	0.924
$n = 100, \sigma^2 = 20$	AIC	0.998
	BIC	0.994

Table 4.1: The proportion of picking the true models with $\sigma^2 = 20$, different sample sizes

	methods	proportion
$n = 5, \sigma^2 = 0.1$	AIC	1
	BIC	1
$n = 5, \sigma^2 = 1$	AIC	0.997
	BIC	0.996
$n = 5, \sigma^2 = 5$	AIC	0.890
	BIC	0.909
$n = 5, \sigma^2 = 10$	AIC	0.747
	BIC	0.801
$n = 5, \sigma^2 = 20$	AIC	0.619
	BIC	0.649

Table 4.2: The proportion of picking the true models with sample size $n = 5$, different σ^2

$(\beta_1, \beta_2, \beta_3)$	(0.2,1,1)		(1,5,5)		(2,10,10)	
σ^2	AIC	BIC	AIC	BIC	AIC	BIC
1	1	1	1	1	1	1
2	1	0.995	1	1	1	1
5	0.732	0.428	1	1	1	1
10	0.326	0.118	1	10.995	1	1
15	0.234	0.060	0.974	0.873	1	1
20	0.171	0.031	0.856	0.643	1	0.997
25	0.098	0.023	0.736	0.429	0.995	0.951

Table 4.3: The proportion of picking the true models with sample size is 100, different σ^2 and parameters

From the appendix B.4, the first four tables compare the proportions of picking the true models in AIC and BIC when variance is fixed and changing sample size. For each table, we can see when the sample size is getting

bigger, the proportion gets bigger and close to 1. In the fourth table, we can see that the proportions for n=5 are bigger than the proportions for n=10, which indicates that there is some inconsistency for a small sample size. The next four tables compare the proportions of picking the true models in AIC and BIC when the sample size is fixed and we change variance. We can easily see that increasing variance will decrease the proportion. The last table compares the proportions of picking the true models in AIC and BIC when the sample size is fixed to 100 with different variances and parameters. We can see that bigger values of parameters will give bigger values of proportions and AIC performs better than BIC. The proportion of choosing the true models in BIC is smaller than the proportion of choosing the true models in AIC with fixed variance, sample size and parameters.

4.4.2 Multivariate linear regression models

In this section, we use the GPS data to estimate the multivariate linear regression models. The data used for this estimation is NumSats are greater than 0 without outliers removed. There are 6 variables in this estimation: 2 dependent variables (response variables): Longitude (X) and Latitude (Y) in degrees, and 4 independent variables (predictor variables): GPSTOW in hours (Z_1), NavType (Z_2), NumSats (Z_3) and Altitude.MSL in meters (Z_4). The number of the observations is 2671. NavType and NumSats are used as factors. AIC and BIC are used to find the best fit model for the data. There are 11 GPS position fix types: 4, 6, 14, 16, 204, 206, 1006, 1013, 1015, 1016, 1206. There are 5 NumSats: 3, 4, 5, 6, 7.

Model 1 is the full model that includes all variables :

$$\begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} \beta_{01} \\ \beta_{02} \end{pmatrix} + \begin{pmatrix} \beta_{11} \\ \beta_{12} \end{pmatrix} Z_1 + \begin{pmatrix} \beta_{21} \\ \beta_{22} \end{pmatrix} Z_2 + \begin{pmatrix} \beta_{31} \\ \beta_{32} \end{pmatrix} Z_3 + \begin{pmatrix} \beta_{41} \\ \beta_{42} \end{pmatrix} Z_4 + \begin{pmatrix} \varepsilon_X \\ \varepsilon_Y \end{pmatrix}$$

AIC for model 1 is :-75155.59, and BIC for model 1 is -75043.67.
Then we use the backward elimination method to test AIC and BIC for the

reduced models to see if there are better fitted models. After the test, we find the model which has the smallest AIC and BIC, it is the best fit model for the data. the model is below:

$$\begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} \beta_{01} \\ \beta_{02} \end{pmatrix} + \begin{pmatrix} \beta_{11} \\ \beta_{12} \end{pmatrix} Z_1 + \begin{pmatrix} \beta_{21} \\ \beta_{22} \end{pmatrix} Z_2 + \begin{pmatrix} \beta_{41} \\ \beta_{42} \end{pmatrix} Z_4 + \begin{pmatrix} \varepsilon_X \\ \varepsilon_Y \end{pmatrix}$$

AIC for this model is: -75155.51, and BIC for this model is: -75063.34.

The summary statistics for the predicted variables in the fitted model are below:

NavType	Frequency	Percentage
4	1194	0.4470
6	104	0.0389
14	198	0.0741
16	10	0.0037
204	766	0.2868
206	11	0.0041
1006	59	0.0221
1013	6	0.0022
1015	183	0.0685
1016	11	0.0041
1206	129	0.0483
Total	2671	1

Table 4.4: Tables of summarised frequency and percentage for the NavType in the fitted model

Variables	Median	Mean	S.D.
GPSTOW (hours)	102.70	102.5	11.68443
AltitudeMSL (meters)	31.01	44.39	153.8322

Table 4.5: Tables of summarised statistics for GPSTOW and AltitudeMSL in the fitted model

After the model selection, we use R to summarise the fitted model to estimate the parameters for the model. The tables for output are below:

Coefficient	Estimate	S.E.	$p - value$
Intercept	1.750×10^{9}	3.630×10^{2}	$< 2 \times 10^{-16}$
GPSTOW	-9.107×10^{-7}	9.777×10^{-7}	0.351707
(NavType) 6	-2.709×10^{2}	2.142×10^{2}	0.205998
(NavType)14	-4.781×10^{2}	1.609×10^{2}	0.002992
(NavType)16	6.139×10^{2}	6.647×10^{2}	0.355797
(NavType)204	36.55	96.93	0.706157
(NaveType)206	2.179×10^{3}	6.341×10^{2}	0.00600
(NavType)1006	-9.358×10^{2}	2.806×10^{2}	0.000865
(NavType)1013	1.785×10^{2}	8.568×10^{2}	0.834951
(NavType)1015	-1.162×10^{2}	1.762×10^{2}	0.509384
(NavType)1016	-7.464×10^{2}	6.340×10^{2}	0.239212
(NavType)1206	2.029×10^{2}	1.944×10^{2}	0.296612
AltitudeMSL	-0.5910	2.784×10^{-3}	$< 2 \times 10^{-16}$

Table 4.6: Tables of summarised variables for Longitude

Coefficient	Estimate	S.E.	$p-value$
Intercept	-4.098×10^8	3.762×10^2	$< 2 \times 10^{-16}$
GPSTOW	3.082×10^{-6}	1.013×10^{-6}	0.00237
(NavType) 6	-4.473×10^2	2.220×10^2	0.04400
(NavType)14	1.409×10^3	1.668×10^2	$< 2 \times 10^{-16}$
(NavType)16	-7.534×10^2	6.888×10^2	0.27415
(NavType)204	1.811×10^2	1.005×10^2	0.07151
(NaveType)206	-4.771×10^2	6.572×10^2	0.46787
(NavType)1006	1.679×10^3	2.908×10^2	8.71×10^{-9}
(NavType)1013	-41.61	8.880×10^2	0.96263
(NavType)1015	3.031×10^2	1.826×10^2	0.09699
(NavType)1016	1.747×10^2	6.571×10^2	0.79037
(NavType)1206	2.901×10^2	2.014×10^2	0.14995
AltitudeMSL	-0.3587	2.885×10^{-3}	$< 2 \times 10^{-16}$

Table 4.7: Tables of summarised variables for Latitude

Variable	D.f.	$p-value$
GPSTOW	2	3.49×10^{-7}
factor(NavType)	20	$< 2 \times 10^{-16}$
AltitudeMSL	2	$< 2 \times 10^{-16}$

Table 4.8: Tables of summarised MANOVA results for model 3

From the Table 4.8, we see that the p-values of the three variables are all very small and indicates we can not reject any of them. It means they all have a relationship with Longitude and Latitude.

4.5 Summary

Since AIC and BIC are the most commonly used methods for the selection of good models, we use AIC and BIC as criteria in this chapter for the linear regression models. We tested whether AIC and BIC are influenced by sample size and variance values and found GPSTOW, NavType and AltitudeMSL are the three variables in the best fitted model for the bivariate response variables; Longitude and Latitude.

Chapter 5

Change Point Detection

In this chapter, we would like to investigate if there are some shifts in the location based on the GPS data we used in previous chapters. We use the Longitude, Latitude and Altitude converted to meters as our three observed variables (after deleting outliers). We separate the observations into 41 groups based on GPSTOW, as the measurement is one hour per group.

Multivariate statistical process control (SPC) carries out ongoing checks to ensure that a process is in-control and to detect when it is out of control. An in-control process is one in which variations in process measurements can be attributed to chance causes. In this study, the process measurements are the location coordinates for the GPS receiver. The chance causes include number and position of satellites, atmospheric conditions which lead to variation in the location coordinates when the receiver has not moved. The aim in this chapter is therefore to investigate methods for detecting changes in location coordinates that can be attributed to movement rather than chance causes.

These checks on the process are traditionally done by T^2, multivariate cumulative sum (CUSUM), and multivariate exponentially weighted moving average control charts. These traditional SPC charts assume that the in-control true parameters are known and use these assumed true val-

ues to set the control limits. But the reality is that true parameter values are seldom, if ever, known exactly. In such cases, the statistical process control procedure is then divided into two phases. In Phase I, historical data is used to compute initial control limits. Any points outside the limits are investigated and possibly discarded, and if necessary the limits are recomputed. In Phase II, the limits and parameter estimates from Phase I are used for real-time monitoring.

The Phase I study needs large samples but some industrial settings have a paucity of relevant data for estimating the process parameters [45]. Zamba and Hawkins (2006) [46] outlined an attractive alternative to traditional charting methods when monitoring for a step change in the mean vector. Their method is based on an unknown-parameter likelihood ratio test for a change in mean of p-variate normal data.

Hawkins and Olwell (1997) also introduced use of Shewhart charting-Hotelling's T^2 for a multivariate quality control problem and the use of CUSUM charting for a smaller but persistent shifts. They proposed that regression adjustment can be helpful in resolving the problem in diagnosing shifts [10].

Pettitt (1979) presented some simple techniques for testing for a change of distribution in a sequence of observations when the initial distribution is unknown. For discrete Bernoulli and Binomial data, exact and conservative tests have been developed which are simple to use. For continuous data, approximate tests are developed which are both simple and robust against changes in distributional form [36].

Lai (1995) introduced Control Charts for Multivariate and Serially Correlated Sample, Statistical Quality Control and Shewhart's Control Charts and Moving Average Charts for Detection of Mean Changes [28].

5.1 Hotelling's T^2 chart

We consider two methods for detecting location shifts. The first method uses T^2 statistics for an observed vector $\mathbf{X}_i$ and calculates the upper control limit (UCL) to see if there is a change for the mean [46]. In our case, $\mathbf{X}_i$ is the mean for each group, i is from 1 to 41 based on the GPSTOW as the measurement for one hour per group and in each group we have three variables: Longitude, Latitude and Altitude.

The T^2 statistic for an observed vector $\mathbf{X}_i$ is given by

$$T_i^2 = (\mathbf{X}_i - \bar{\mathbf{X}})' S^{-1} (\mathbf{X}_i - \bar{\mathbf{X}}).$$

The T_i^2 statistic has a chi-squared distribution with p degrees of freedom. $\bar{\mathbf{X}} = \frac{1}{n}\sum_{j=1}^{n} \mathbf{X}_j$ is a mean vector, and $S = \frac{1}{n-1}\sum_{j=1}^{n}(\mathbf{X}_i - \bar{\mathbf{X}})(\mathbf{X}_i - \bar{\mathbf{X}})'$ is a covariance matrix. $\bar{\mathbf{X}}$ and S are determined from historical data to estimate the true mean $\boldsymbol{\mu}$ and covariance matrix Σ if these are unknown. In our case, $\bar{\mathbf{X}}$ and S are the mean and covariance matrix for all 41 groups. At each time point i, T_i^2 is used to test the hypothesis:

$$\mathrm{H}_0 : \boldsymbol{\mu}_i = \boldsymbol{\mu}$$

$$\mathrm{H}_1 : \boldsymbol{\mu}_i \neq \boldsymbol{\mu}$$

$T_i^2 \in [0, \infty)$, so even though we have a two-sided alternative hypothesis, we can only detect differences, but not the direction of the difference.

The UCL is calculated as:

$$UCL = \frac{(n-1)(n+1)p}{n(n-p)} F_{\alpha,p,n-p}$$

where $F_{\alpha,p,n-p}$ is the upper αth percentile point of the F distribution with $(p, n-p)$ degrees of freedom. In here, α is 0.05 and p is 3. The plot shows T^2 and UCL below, the value of UCL is 1.473646.

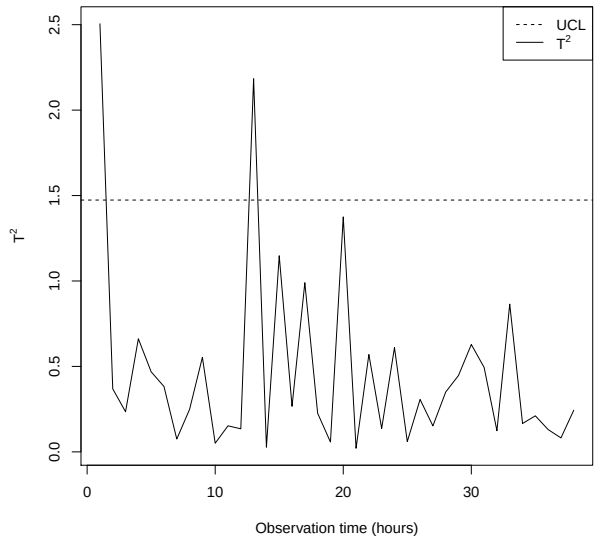


Figure 5.1: Values of T^2 method in observation time with UCL

5.2 Multivariate exponentially weighted moving average

The second method is the multivariate exponentially weighted moving average (MEWMA). It is due to Lowry et al. (1992) [30], and is based on the recursion

$$\mathbf{M}_0 = \boldsymbol{\mu},$$
$$\mathbf{M}_i = \gamma \mathbf{X}_i + (1-\gamma)\mathbf{M}_{i-1}.$$

A shift is signalled if $C_i = (\mathbf{M}_i - \boldsymbol{\mu})'\Sigma^{-1}(\mathbf{M}_i - \boldsymbol{\mu}) > h$, where $h(>0)$ is a threshold set according to a desired false-alarm rate, the value of h is chosen to achieve a specified in-control average run length (ARL). The parameters $\boldsymbol{\mu}$ and Σ denote the true mean and covariance matrix for the data, respectively. $\mathbf{M}_i$ is a smoothed estimate of the current mean vector and γ is the smoothing constant lying between 0 and 1. This chart is sen-

sitive to small persistent shifts in any direction, with small values of the tuning constant γ used to detect small shifts and larger values used for larger shifts [30].

5.2.1 Determine the value of h

In order to find h, we need to decide on the in-control ARL first. The threshold value, h, is largely determined by simulation [10]. We choose an in-control ARL of about a year, or approximately 8760 hours. This means our MEWMA will signal a change in mean once a year, on average, when there has been no shift. In other words we will have one false alarm once a year. That is sensible, since investigation of vertical land movement around the New Zealand coastline showed that the relative sea level change in 2004 in Wellington was -0.34mm [4]. Therefore in the absence of a landslide, land movement is minimal.

We set γ to be 0.4 since we want to detect small shifts. The time frame is 10000 hours, so we have 10000 mean vectors. In each hour, we simulate 50 data points from a normal distribution with the mean vector μ and covariance matrix the same as those for the observed data with outliers removed. For a given value of h, we ran the simulation 100 times, and each time we got a run length. We then calculated the ARL for the 100 simulations. The ARL is calculated by determining the median of the 100 run lengths rather than mean of the 100 run lengths, and this is because the run lengths are skewed [22].

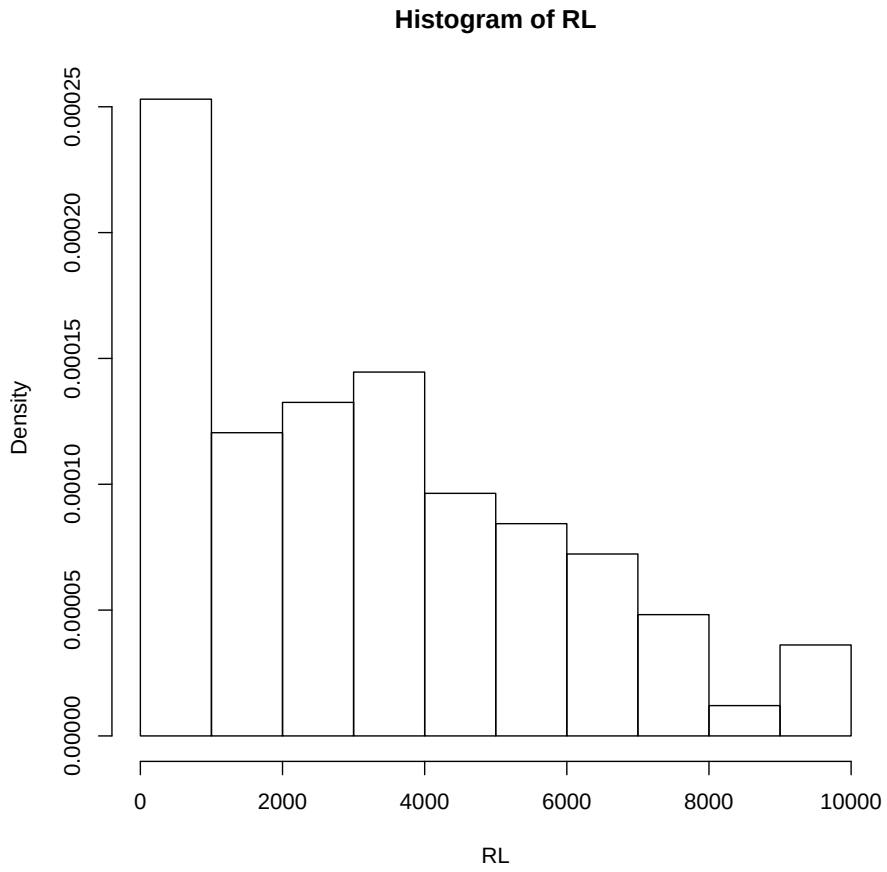
The table below shows different h values and their corresponding ARL values.

Table 5.1: Simulated ARL and h

h	0.065	0.08	0.095	0.1	0.1068
ARL (in hours)	168	720	2160	4320	8760
Frequency of false alarm	Weekly	Monthly	3 months	6 months	1year

5.2.2 Results

The histogram below shows the frequency of the run length values, because there are some infinite values in the run lengths and they can not appear in the histogram, so we choose the median to be the ARL.



We tried different values of h to get an ARL close to a year. We found that when h is 0.1068, that the ARL is 8224 hours (close to a year), therefore we choose 0.1068 as our h value. Now we have found h, we can use $h = 0.1068$ in our observations, and then plot C_i and h together. The plot is below:

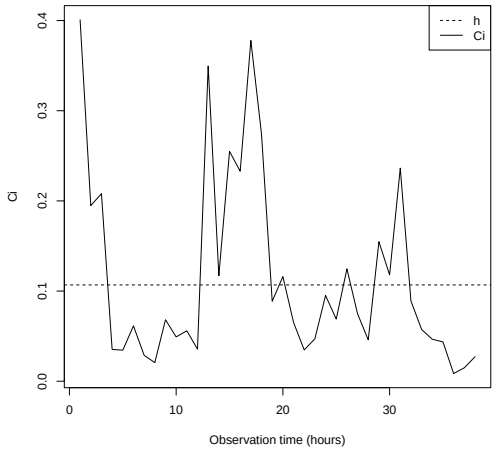


Figure 5.2: Observation times and C_i with h

5.2.3 Table of results of h for different values of γ and ARL

The h value in Figure 5.2 is small, and is influenced by the value of γ, so we want to test different γ values and see the results of h. The table below shows that when γ increases, the corresponding h increases.

Frequency of false alarm	Weekly	Monthly	3 months	6 months	1year
ARL(in hours)	168	720	2160	4320	8760
$\gamma = 0.2$	0.027	0.0355	0.04	0.043	0.047
$\gamma = 0.4$	0.065	0.08	0.095	0.1	0.1068
$\gamma = 0.6$	0.115	0.142	0.16	0.18	0.186
$\gamma = 0.8$	0.18	0.221	0.25	0.265	0.285
$\gamma = 1$	0.265	0.32	0.373	0.4	0.435

Table 5.2: Results of h in different values of γ and ARL

5.3 Discussion

From Figure 5.1, we can see that most of the T^2 values are below the UCL and only two T^2 are above, that suggests a special cause, and will require more investigation. From Figure 5.2, we can see that the h value is too small and more than half of the points C_i are above the h line, that is because in the observations, the data is not normally distributed and also is influenced by the chosen value of γ. From Table 5.2, we can see that if we choose γ to be 1 and ARL to be 1 year, h will be 0.435, which is greater than all the values of points C_i.

Chapter 6

Discussion

The main goal of this thesis was to use GPS data to estimate the location of a GPS device and investigate methods for detecting movement of the device.

In chapter 2, the first part is to estimate the position of a GPS device using inaccurate Longitude and Latitude measurements. As there were outliers in the data set, we created two methods for deleting the outliers: the Euclidean distance method and the Mahalanobis distance method. The difference between the two distance methods is that the Mahalanobis distance takes into account the correlations of the data set and is scale-invariant. For our data set, Longitude and Latitude are correlated, hence the Mahalanobis distance method is more suitable for our data. After deleting outliers, we use maximum likelihood estimation to estimate the means and variance-covariance matrix for the new data set based on the assumption that the data is normally distributed.

After we test for normality, we find the data is not normal distributed, therefore we use the bootstrap method to estimate the position and construct a confidence ellipse for the bootstrap sample means. From the results, the mean for the data and bootstrap sample mean are the same, this implies we get the same estimated position from the two methods. Three methods for constructing confidence regions are used. One is to construct

a confidence ellipse based on the sample means and sample covariance matrix, the other two methods construct the simultaneous confidence intervals (SCI) based on the percentile bootstrap approach. By comparing the three methods, the confidence ellipse presents a better result for constructing a confidence region for the data.

Second, we determined that the position measurements (i.e. Longitude and Latitude) are dependent on other factors (i.e. GPSTOW, NavType and Altitude.MSL) by using AIC and BIC for the different fitted linear models and finding the smallest values of AIC and BIC (i.e. the best fitted model for the data). Some simulations in different sample sizes and variances for univariate linear regression models are used here to test the reliability and stability of AIC and BIC. The results show that when the sample size and variance are big, AIC and BIC are reliable and stable for choosing the right linear regression models.

We could use spatial regression models to apply to our data, because our data is GPS data. The spatial data is not generally independent, so that statistical inference in ordinary regression models applied to spatial data is suspect, a number of attempts have been made to provide a regression framework in which spatial dependency is taken into account. These approches may generally be decribed as spatial regression models [13].

After the estimation of location, we use two methods to detect the change point for the location. One method is by using T^2 statistics for the observations and calculating the upper control limit to see if there is a change for the mean. The other method is the multivariate exponentially weighted moving average. The results from the two methods both show that there are movements or change points during the observation time interval. This is not correct, because the device was not moved in the time during which data was collected. The reason maybe because of the choice of $\gamma = 0.4$ results in too many false positive signals.

Appendix A

Glossary

C/A-code : C/A-code is coarse acquisition code, it is one of the two GPS codes. Each code consists of a stream of binary digits, zeros and ones, known as bits or chips [12]. The C/A code is a pseudo-random code (PRN) which looks like a random code but is clearly defined for each satellite. It is repeated every 1023 bits or every millisecond. Therefore each second 1023000 chips are generated. Taking into account the speed of light the length of one chip can be calculated to be 300 m [25].

drms : Distance Root Mean Squared. DRMS is a single number that express 2D accuracy. In order to compute the DRMS of horizontal position errors, the standard errors (σ) from the known position in the directions of the coordinate axis are required. DRMS is the square root of the average of the square errors which is defined as: $\mathrm{DRMS} = \sqrt{\sigma_x^2 + \sigma_y^2}$ [41].

selective availability : Selective availability (SA) was an intentional degradation of public GPS signals implemented for national security reasons. In May 2000, at the direction of President Bill Clinton, the U.S government discontinued its use of Selective Availability in order to make GPS more responsive to civil and commercial users worldwide

[33].

Carrier-phase measurement : the carrier phase measurement is a measure of the range between a satellite and receiver expressed in units of cycles of the carrier frequency. This measurement can be made with very high precision (of the order of millimetres), but the whole number of cycles between satellite and receiver is not measurable [35].

pseudorange measurement : Time that the signal is transmitted from the satellite is encoded on the signal, using the time according to an atomic clock onboard the satellite. Time of signal reception is recorded by receiver using an atomic clock. A receiver measures difference in these times: $\mathrm{pseudorange} = (\mathrm{timedifference}) \times (\mathrm{speedoflight})$. Pseudorange is almost like range, except that it includes clock errors because the receiver clocks are far from perfect [6].

ephemeris or orbital errors : The satellite ephemeris bias is the discrepancy between the true position (and velocity) of a satellite and its known value. This discrepancy can be parameterised in a number of ways, but a common way is via the three orbit components: alongtrack, crosstrack and radial. Orbit error is a residual bias, arising from mismodelling of the satellite trajectory, or accepting as "true" an ephemeris that has errors. In the case of the Broadcast Ephemerides within the GPS Navigation Message, these errors can range from (usually) less then 10m to (very rarely) up to 100m [38].

Multipath error : Errors caused by the interference of a signal that has reached the receiver antenna by two or more different paths. This is usually caused by one path being bounced or reflected. The impact on a pseudo-range measurement may be up to a few metres. In the case of carrier phase, this is of the order of a few centimetres [38].

ionospheric layer : The ionosphere is that region of space containing electrically charged species. The ionosphere is the layer of the atmosphere from 50 to 500 km that consists of ionized air.

tropospheric layer : The troposphere is the lowest portion of Earth's atmosphere.

antenna phase center : Antenna phase center is the electronic center of the antenna. It often does not correspond to the physical center of the antenna. The radio signal is measured at the Antenna Phase Center.

receiver noise : In a GPS receiver the noise translates into errors in range measurement.

satellites clock error : The built in clock of the GPS receiver is not as accurate as the atomic clocks of the satellites and the slight timing errors leads to corresponding errors in calculations.

receiver clock error : The receiver clock is synchronised to GPS through the normal operation of code-correlating receivers to about 0.1 msec accuracy under SA. Therefore residual biases of the order of a dekametre (tens of metres) remain, and must be accounted for in some way.

Appendix B

R code and tables

B.1 R code for chapter 2

```
#This is Euclidean distance method

pkts<-read.table("bin_pkts.txt",header=T, sep="\t");str(pkts)
pkts1<-pkts[pkts$NumSats>0,]; str(pkts1)
T<-pkts1$GPSTOW/(1000*60*60)#in hours
lo<-pkts1$Longitude/10^4
la<-pkts1$Latitude/10^4
lonew<-lo-min(lo)
lanew<-la-min(la)
plot(lonew,lanew)
m<- c(numeric(0),numeric(0))
for (i in 1:1000) {
m<-rbind(m,c(mean(lonew),mean(lanew)))
d<-sqrt((lonew-mean(lonew))^2+(lanew-mean(lanew))^2)
quantile(sort(d),probs=0.99)
nd <- cbind(lonew,lanew,T,d)
      d1 <- subset(nd, nd[,4]<quantile(sort(d),probs=0.99))
      lonew<-d1[,1]
```

```
      lanew<-d1[,2]
      T <- d1[,3]
err <- sqrt((mean(lonew)-m[i,1])^2+(mean(lanew)-m[i,2])^2)
if (err<0.0001){
m<-rbind(m,c(mean(lonew),mean(lanew)))
break
}
}

#this is mahalanobis distance method
pkts<-read.table("bin_pkts.txt",header=T, sep="\t");str(pkts)
pkts1<-pkts[pkts$NumSats>0,]; str(pkts1)
T<-pkts1$GPSTOW/(1000*60*60)#in hours
lo<-pkts1$Longitude/10^4
la<-pkts1$Latitude/10^4
lonew<-lo-min(lo)
lanew<-la-min(la)
set<-cbind(lonew,lanew)
cov(set)
lo<-lonew-mean(lonew)
la<-lanew-mean(lanew)
lola<-cbind(lo,la)
m<- c(numeric(0),numeric(0),numeric(0))
for (i in 1:1000) {
m<-rbind(m,c(mean(lonew),mean(lanew)))
lo<-lonew-mean(lonew)
la<-lanew-mean(lanew)

d<-numeric(0)
for(j in 1:length(lonew)){
distance<-c(lo[j],la[j])%*%(solve(cov(cbind(lonew,lanew))))%*%
```

```
c(t(lo[j]),t(la[j]))
d<-c(d,distance)
}
nd <- cbind(lonew,lanew,T,d)
      d1 <- subset(nd, nd[,4]<quantile(sort(d),probs=0.99))
      lonew<-d1[,1]
      lanew<-d1[,2]
      T <- d1[,3]

err <- sqrt((mean(lonew)-m[i,1])^2+(mean(lanew)-m[i,2])^2)
if (err<0.0001){
m<-rbind(m,c(mean(lonew),mean(lanew)))
break
}
}

#################################################################
#Simulations about outliers in three situations and using the two
methods to delete outliers
library(mvtnorm)
# let the outliers far from data
sigma <- matrix(c(3,1,1,4), ncol=2)
x1<- rmvnorm(n=9800, mean=c(5,10), sigma=sigma,method="chol")
colMeans(x1)
var(x1)
plot(x1,ylim=c(-5,20),xlim=c(-5,15),xlab="x",ylab="y",pch=20,
main="Outliers out of the data")
x2<-rmvnorm(n=100,mean=c(0,0),sigma=sigma,method="chol")
x3<-rmvnorm(n=100,mean=c(10,0),sigma=sigma,method="chol")
points(x2,col="red",pch=19)
points(x3,col="green",pch=19)
```

```
x<-rbind(x1,x2,x3)
points(mean(x[,1]),mean(x[,2]),col="yellow",pch=20)
colMeans(x)
# let the outliers inside the data
sigma <- matrix(c(3,1,1,4), ncol=2)
x1<- rmvnorm(n=9800, mean=c(5,10), sigma=sigma,method="chol")
colMeans(x1)
var(x1)
plot(x1,ylim=c(-5,20),xlim=c(-5,15),xlab="x",ylab="y",
main="Outliers inside the data",pch=20)
x2<-rmvnorm(n=100,mean=c(2,7),sigma=sigma,method="chol")
x3<-rmvnorm(n=100,mean=c(8,7),sigma=sigma,method="chol")
points(x2,col="red",pch=19)
points(x3,col="green",pch=19)
x<-rbind(x1,x2,x3)
#points(mean(x[,1]),mean(x[,2]),col="yellow",pch=20)


# let the outliers near the data
sigma <- matrix(c(3,1,1,4), ncol=2)
x1<- rmvnorm(n=9800, mean=c(5,10), sigma=sigma,method="chol")
colMeans(x1)
var(x1)
plot(x1,ylim=c(-5,20),xlim=c(-10,20),xlab="x",ylab="y",
main="Outliers near the data",pch=20)
x2<-rmvnorm(n=100,mean=c(-2,10),sigma=sigma,method="chol")
x3<-rmvnorm(n=100,mean=c(12,10),sigma=sigma,method="chol")
points(x2,col="red",pch=19)
points(x3,col="green",pch=19)
x<-rbind(x1,x2,x3)
```

```
#Euclidean distance
x<-rbind(x1,x2,x3)
y<-c(rep(1,9800),rep(2,100),rep(3,100))
z<-cbind(x,y)

z1<-z[,1]
z2<-z[,2]
z3<-z[,3]
d<-sqrt((z1-mean(z1))^2+(z2-mean(z2))^2)

m<- c(numeric(0),numeric(0))
for (i in 1:10000) {
m<-rbind(m,c(mean(z1),mean(z2)))
d<-sqrt((z1-mean(z1))^2+(z2-mean(z2))^2)
quantile(sort(d),probs=0.99)
nd <- cbind(z1,z2,z3,d)
      d1 <- subset(nd, nd[,4]<quantile(sort(d),probs=0.99))
      z1<-d1[,1]
      z2<-d1[,2]
      z3<- d1[,3]
err <- sqrt((mean(z1)-m[i,1])^2+(mean(z2)-m[i,2])^2)
if (err<0.001){
m<-rbind(m,c(mean(z1),mean(z2)))
break
}
}
#Mahalanobis distance
y<-c(rep(1,9800),rep(2,100),rep(3,100))
z<-cbind(x,y)
z1<-z[,1]
z2<-z[,2]
```

```
z3<-z[,3]

m<- c(numeric(0),numeric(0))
for (i in 1:10000) {
m<-rbind(m,c(mean(z1),mean(z2)))
z1d<-z1-mean(z1)
z2d<-z2-mean(z2)
d<-numeric(0)
for(j in 1:length(z1)){
distance<-c(z1d[j],z2d[j])%*%(solve(cov(cbind(z1,z2))))%*%
c(t(z1d[j]),t(z2d[j]))
d<-c(d,distance)
}
nd <- cbind(z1,z2,z3,d)
      d1 <- subset(nd, nd[,4]<quantile(sort(d),probs=0.99))
      z1<-d1[,1]
      z2<-d1[,2]
      z3<- d1[,3]
err <- sqrt((mean(z1)-m[i,1])^2+(mean(z2)-m[i,2])^2)
if (err<0.001){
m<-rbind(m,c(mean(z1),mean(z2)))
break
}
}
```

B.2 R code for chapter 3

```
#bootstrap for means(Mahalanobis)
pkts<-read.table("bin_pkts.txt",header=T, sep="\t");str(pkts)
```

```
pkts1<-pkts[pkts$NumSats>0,]; str(pkts1)
T<-pkts1$GPSTOW/(1000*60*60)#in hours
lo<-pkts1$Longitude/10^4
la<-pkts1$Latitude/10^4
lonew<-lo-min(lo)
lanew<-la-min(la)
set<-cbind(lonew,lanew)
cov(set)
lo<-lonew-mean(lonew)
la<-lanew-mean(lanew)
lola<-cbind(lo,la)
m<- c(numeric(0),numeric(0),numeric(0))
for (i in 1:1000) {
m<-rbind(m,c(mean(lonew),mean(lanew)))
lo<-lonew-mean(lonew)
la<-lanew-mean(lanew)

d<-numeric(0)
for(j in 1:length(lonew)){
distance<-c(lo[j],la[j])%*%(solve(cov(cbind(lonew,lanew))))%*%
c(t(lo[j]),t(la[j]))
d<-c(d,distance)
}
nd <- cbind(lonew,lanew,T,d)
      d1 <- subset(nd, nd[,4]<quantile(sort(d),probs=0.99))
      lonew<-d1[,1]
      lanew<-d1[,2]
      T <- d1[,3]

err <- sqrt((mean(lonew)-m[i,1])^2+(mean(lanew)-m[i,2])^2)
if (err<0.0001){
```

```
m<-rbind(m,c(mean(lonew),mean(lanew)))
break
}
}
x<-nd[,1]
y<-nd[,2]

library(mixtools)
library(mvtnorm)

ellipse(mu=c(mean(x),mean(y)),sigma=cov(cbind(x,y)), alpha = .05,
npoints = 100)
id<-c(1:length(x))
data_or<-data.frame(id,x,y)
B<-1000
meanvec<-matrix(rep(0,2*B),nrow=B)

for(b in 1:B){
sampleb<-sample(data_or$id,length(x),replace=TRUE)
meanvec[b,1]<-mean(data_or[sampleb,2])
meanvec[b,2]<-mean(data_or[sampleb,3])

}
meanvec
cov(meanvec)
points(meanvec,col="red",pch=19)
ellipse(mu=colMeans(meanvec),sigma=cov(meanvec), alpha = .05,
npoints = 100,col="yellow")

#bootstrap confidence ellipse for means (Euclidean)
pkts<-read.table("bin_pkts.txt",header=T, sep="\t");str(pkts)
```

```
pkts1<-pkts[pkts$NumSats>0,]; str(pkts1)
T<-pkts1$GPSTOW/(1000*60*60)#in hours
lo<-pkts1$Longitude/10^4
la<-pkts1$Latitude/10^4
lonew<-lo-min(lo)
lanew<-la-min(la)

m<- c(numeric(0),numeric(0))
for (i in 1:1000) {
m<-rbind(m,c(mean(lonew),mean(lanew)))
d<-sqrt((lonew-mean(lonew))^2+(lanew-mean(lanew))^2)
quantile(sort(d),probs=0.99)
nd <- cbind(lonew,lanew,T,d)
      d1 <- subset(nd, nd[,4]<quantile(sort(d),probs=0.99))
      lonew<-d1[,1]
      lanew<-d1[,2]
      T <- d1[,3]
err <- sqrt((mean(lonew)-m[i,1])^2+(mean(lanew)-m[i,2])^2)
if (err<0.0001){
m<-rbind(m,c(mean(lonew),mean(lanew)))
break
}
}
x<-nd[,1]
y<-nd[,2]
plot(x,y)
library(mixtools)
library(mvtnorm)

ellipse(mu=c(mean(x),mean(y)),sigma=cov(cbind(x,y)), alpha = .05,
npoints = 100)
```

```
id<-c(1:length(x))
data_or<-data.frame(id,x,y)
B<-1000
meanvec<-matrix(rep(0,2*B),nrow=B)

for(b in 1:B){
sampleb<-sample(data_or$id,length(x),replace=TRUE)
meanvec[b,1]<-mean(data_or[sampleb,2])
meanvec[b,2]<-mean(data_or[sampleb,3])

}
meanvec
cov(meanvec)
points(meanvec,col="red",pch=19)
ellipse(mu=colMeans(meanvec),sigma=cov(meanvec), alpha = .05,
npoints = 100,col="yellow")

#use algoritm 1 to find the limits

id<-c(1:length(x))
data_or<-data.frame(id,x,y)
B<-1000
meanvec<-matrix(rep(0,2*B),nrow=B)

for(b in 1:B){
sampleb<-sample(data_or$id,length(x),replace=TRUE)
meanvec[b,1]<-mean(data_or[sampleb,2])
meanvec[b,2]<-mean(data_or[sampleb,3])

}
x<-meanvec[,1]
```

```
y<-meanvec[,2]
rank.x<-rank(x)
rank.y<-rank(y)
rank<-cbind(rank.x,rank.y)
r<-numeric(0)
for(i in 1:1000){
rank.max<-max(rank[i,])
r<-c(r,rank.max)
}
r
all<-cbind(x,y,rank.x,rank.y,r)
sort.all<-all[order(all[,5]),]
a<-sort.all[length(x)*0.975,1]
b<-sort.all[length(x)*0.975,2]
r<-numeric(0)
for(i in 1:length(x)){
rank.min<-min(rank[i,])
r<-c(r,rank.min)
}

all<-cbind(x,y,rank.x,rank.y,r)
sort.all<-all[order(all[,5]),]
c<-sort.all[length(x)*0.025,1]
d<-sort.all[length(x)*0.025,2]
plot(meanvec,xlab="Bootstrap mean of latitude",
ylab="bootstrap mean of longitude",col="red",pch=19)
rect(c,d,a,b , density = NULL, angle = 90,
     col =NA, border ="blue")

#use algorithm 2 to find the limits
```

```
id<-c(1:length(x))
data_or<-data.frame(id,x,y)
B<-1000
meanvec<-matrix(rep(0,2*B),nrow=B)

for(b in 1:B){
sampleb<-sample(data_or$id,length(x),replace=TRUE)
meanvec[b,1]<-mean(data_or[sampleb,2])
meanvec[b,2]<-mean(data_or[sampleb,3])

}
x<-meanvec[,1]
y<-meanvec[,2]
rank.x<-rank(x)
rank.y<-rank(y)
#step 2
rank.x.star<-abs(rank.x-1001/2)
sign.x<-sign(rank.x-1001/2)
rank.y.star<-abs(rank.y-1001/2)
sign.y<-sign(rank.y-1001/2)
#step 3
rank.star<-cbind(rank.x.star,rank.y.star,sign.x,sign.y)
r.star<-r.star.col<-r.star.sign<-c(rep(0,1000))

r.star.mat<-cbind(r.star,r.star.col,r.star.sign)
for(i in 1:1000){
r.star.mat[i,1]<-a<-max(rank.star[i,1:2])
r.star.mat[i,2]<-ifelse((a==rank.star[i,1]),1,2)
r.star.mat[i,3]<-rank.star[i,r.star.mat[i,2]+2]
}
```

```
#step 4
all<-cbind(rank.star,r.star.mat)
r.b<-1001/2+r.star.mat[,1]*r.star.mat[,3]
#step 5
all<-cbind(x,y,rank.x,rank.y,r.b)
new.rank.x<-r.b
new.rank.y<-r.b
all<-cbind(x,y,new.rank.x)
sort.all<-all[order(all[,3]),]
e<-sort.all[length(x)*0.975,1]
b<-sort.all[length(x)*0.975,2]
for(i in 1:length(x)){
r.star.mat[i,1]<-a<-min(rank.star[i,1:2])
r.star.mat[i,2]<-ifelse((a==rank.star[i,1]),1,2)
r.star.mat[i,3]<-rank.star[i,r.star.mat[i,2]+2]
}
all<-cbind(rank.star,r.star.mat)
r.b<-(length(x)+1)/2+r.star.mat[,1]*r.star.mat[,3]
#step 5
all<-cbind(x,y,rank.x,rank.y,r.b)
new.rank.x<-r.b
new.rank.y<-r.b
all<-cbind(x,y,new.rank.x)
sort.all<-all[order(all[,3]),]
c<-sort.all[length(x)*0.025,1]
d<-sort.all[length(x)*0.025,2]
plot(meanvec,xlab="Bootstrap mean of latitude",
ylab="bootstrap mean of longitude",pch=19,col="red")
rect(c,d,e,b, density = NULL, angle = 90,
     col =NA, border = "blue")
```

B.3 R code for chapter 4

```
#Simulation of AIC and BIC with different sample size and variance in
different models.
> alpha<-0.5
> beta1<-2
> beta2<-10
> beta3<-10
>
> samplesize <-c(5,10,20,30,40,50,60,70,80,90,100)
> variance <-c(0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1,2,5)
>
> for (j in 1:length(variance)){
+    currentvariance=variance[j]
+ print(currentvariance)
+ for (i in 1:length(samplesize)) {
+       currentsize = samplesize[i]
+ print(currentsize)
+ x1<-rnorm(currentsize,5,5)
+ x1
+ x2<-rnorm(currentsize,10,5)
+ x2
+ x3<-rnorm(currentsize,20,5)
+ x3
+ y<-rnorm(currentsize,alpha+beta1*x1+beta2*x2+beta3*x3,
currentvariance)
+ y
+ fit1<-lm(y~x1+x2+x3)
+ f1 = c(AIC(fit1),BIC(fit1))
```

```
+ print(f1)
+ fit2<-lm(y~x1+x2)
+ f2 = c(AIC(fit2),BIC(fit2))
+ print(f2)
+ fit3<-lm(y~x2+x3)
+ f3 = c(AIC(fit3),BIC(fit3))
+ print(f3)
+ fit4<-lm(y~x1+x3)
+ f4 = c(AIC(fit4),BIC(fit4))
+ print(f4)
+      fit5<-lm(y~x1)
+ f5= c(AIC(fit5),BIC(fit5))
+ print(f5)
+ fit6<-lm(y~x2)
+ f6 = c(AIC(fit6),BIC(fit6))
+ print(f6)
+ fit7<-lm(y~x3)
+ f7 = c(AIC(fit7),BIC(fit7))
+ print(f7)
+ }
+
+ }


#compare AIC and BIC in different sample size
alpha<-0.5
beta1<-2
beta2<-10
beta3<-10

samplesize <-c(5,10,20,30,40,50,60,70,80,90,100)
```

```
variance <- c(0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1,2,5)
par(mfrow=c(3,4))

for (j in 1:length(variance)){
    currentvariance=variance[j]
AICv <-c(0,0,0,0,0,0,0,0,0,0)
BICv <-c(0,0,0,0,0,0,0,0,0,0)
for (i in 1:length(samplesize)) {
        currentsize = samplesize[i]
print(currentsize)
x1<-rnorm(currentsize,5,5)
x2<-rnorm(currentsize,10,5)
x3<-rnorm(currentsize,20,5)

y<-rnorm(currentsize,alpha+2*x1+10*x2+10*x3,currentvariance)
fit1<-lm(y~x1+x2+x3)
AICv[i] = AIC(fit1)
BICv[i] = BIC(fit1)

}
plot(AICv~samplesize,type="o",col="blue",xlab="sample size",
ylab="criterion",main=paste(c("variance=",currentvariance),
collapse=""),pch=19)

lines(BICv~samplesize, type="o", pch=19, lty=2, col="red")

}
#compare AIC and BIC in different sd
samplesize <-c(5,10,20,30,40,50,60,70,80,90,100)
variance <- c(0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1,2,5)
par(mfrow=c(3,4))
```

```
for (j in 1:length(samplesize)){
    currentsize=samplesize[j]
AICv <-c(0,0,0,0,0,0)
BICv <-c(0,0,0,0,0,0)
for (i in 1:length(variance)) {
        currentvariance = variance[i]
x1<-rnorm(currentsize,5,5)
x2<-rnorm(currentsize,10,5)
x3<-rnorm(currentsize,20,5)
y<-rnorm(currentsize,0.5+2*x1+10*x2+10*x3,currentvariance)
fit1<-lm(y~x1+x2+x3)
AICv[i] = AIC(fit1)
BICv[i] = BIC(fit1)
}

plot(variance,AICv,type="o",col="blue",xlab="variance",
ylab="criterion",main=paste(c("n=",currentsize),collapse=""),
pch=19,lty=1)

lines(BICv~variance, type="o", pch=19, lty=2, col="red")
#pch=19 mean solid circle


}
# plot AIC and BIC vs variance/sample size.
alpha<-0.5
beta1<-2
beta2<-10
beta3<-10
```

```
samplesize <-c(5,10,20,30,40,50,60,70,80,90,100)
variance <- c(0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1,2,5,10,20)
A <- numeric(0)
B <- numeric(0)
for (j in 1:length(variance)){
    currentvariance=variance[j]
print(currentvariance)
for (i in 1:length(samplesize)) {
        currentsize = samplesize[i]
print(currentsize)
x1<-rnorm(currentsize,5,5)
x1
x2<-rnorm(currentsize,10,5)
x2
x3<-rnorm(currentsize,20,5)
x3

y<-rnorm(currentsize,alpha+beta1*x1+beta2*x2+beta3*x3,currentvariance)
y
fit1<-lm(y~x1+x2+x3)
f1 = AIC(fit1)
f2=BIC(fit1)
f1[1]
f2[1]
A <- c(A,f1[1])

B<-c(B,f2[1])

}
```

```
}
A
B
samplesize<-rep(c(5,10,20,30,40,50,60,70,80,90,100),14)
samplesize
variance<-c(rep(0.1,11),rep(0.2,11),rep(0.3,11),rep(0.4,11),
rep(0.5,11),rep(0.6,11),rep(0.7,11),rep(0.8,11),
rep(0.9,11),rep(1,11),rep(2,11),rep(5,11),rep(10,11),rep(20,11))
variance
VS<-variance/samplesize
VS
plot(B~VS,xlab="Variance/Sample size",ylab="BIC")
plot(A~VS,xlab="Variance/Sample size",ylab="AIC")
# 3d scatterplots for AIC and BIC with different sample sizes and
 variances.
alpha<-0.5
beta1<-2
beta2<-10
beta3<-10

samplesize <-c(5,10,20,30,40,50,60,70,80,90,100)
variance <- c(0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1,2,5,10,20)
A <- numeric(0)
B <- numeric(0)
for (j in 1:length(variance)){
    currentvariance=variance[j]
print(currentvariance)
for (i in 1:length(samplesize)) {
        currentsize = samplesize[i]
print(currentsize)
x1<-rnorm(currentsize,5,5)
```

```
x1
x2<-rnorm(currentsize,10,5)
x2
x3<-rnorm(currentsize,20,5)
x3

y<-rnorm(currentsize,alpha+beta1*x1+beta2*x2+beta3*x3,currentvariance)
y
fit1<-lm(y~x1+x2+x3)
f1 = AIC(fit1)
f2=BIC(fit1)
f1[1]
f2[1]
A <- c(A,f1[1])
B<-c(B,f2[1])
}
}
A
B
samplesize<-rep(c(5,10,20,30,40,50,60,70,80,90,100),14)
samplesize
variance<-c(rep(0.1,11),rep(0.2,11),rep(0.3,11),rep(0.4,11),
rep(0.5,11),rep(0.6,11),rep(0.7,11),rep(0.8,11),
rep(0.9,11),rep(1,11),rep(2,11),rep(5,11),rep(10,11),rep(20,11))
variance

library(scatterplot3d)
scatterplot3d(samplesize,variance,A, zlim=c(-800,800),
main="3D Scatterplot",xlab="Sample size",ylab="Variance",
zlab="AIC")
scatterplot3d(samplesize,variance,B, zlim=c(-800,800),
```

```
main="3D Scatterplot",xlab="Sample size",ylab="Variance",
zlab="BIC")

library(rgl)

plot3d(samplesize, variance, A, col="red", size=2)

#simulation for proportion of picking the true models in different
variance and sample size with fixed parameters betas.
alpha<-0.5
beta1<-2
beta2<-10
beta3<-10

samplesize <-c(5,10,20,30,40,50,60,70,80,90,100)
variance <- c(0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1,2,5,10,20)
numberOfRuns <- 1000.0
for (j in 1:length(variance)){
    currentvariance=variance[j]
for (i in 1:length(samplesize)) {
        currentsize = samplesize[i]
count <- 0
for (k in 1:numberOfRuns){
x1<-rnorm(currentsize,5,5)
x2<-rnorm(currentsize,10,5)
x3<-rnorm(currentsize,20,5)
y<-rnorm(currentsize,alpha+2*x1+10*x2+10*x3,currentvariance)
fit1<-lm(y~x1+x2+x3)
f1 = BIC(fit1)
fit2<-lm(y~x1+x2)
f2 = BIC(fit2)
```

```
fit3<-lm(y~x2+x3)
f3 = BIC(fit3)
fit4<-lm(y~x1+x3)
f4 = BIC(fit4)
      fit5<-lm(y~x1)
f5= BIC(fit5)
fit6<-lm(y~x2)
f6 = BIC(fit6)
fit7<-lm(y~x3)
f7 = BIC(fit7)
if(f1==min(c(f1,f2,f3,f4,f5,f6,f7))){
count <- count +1
}

}
print(sprintf("variance: %f sample size: %d proportion: %f",
currentvariance,currentsize,count/numberOfRuns))
}
}
#simulation for proportion of picking the true models in different
variance and parameters with fixed sample size.
alpha<-0.5
beta1<-c(0.2,0.5,1,2)
beta2<-c(1,2,5,10)
beta3<-c(1,2,5,10)

samplesize <-100
variance <- c(0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1,2,5,10,20)
numberOfRuns <- 1000.0
for (j in 1:length(variance)){
    currentvariance=variance[j]
```

```
for (k in 1:3) {
currentk=c(beta1[k],beta2[k],beta3[k])

count <- 0
for (k in 1:numberOfRuns){
x1<-rnorm(100,5,5)
x2<-rnorm(100,10,5)
x3<-rnorm(100,20,5)
y<-rnorm(100,alpha+currentk[1]*x1+currentk[2]*x2+currentk[3]*x3,
currentvariance)
fit1<-lm(y~x1+x2+x3)
f1 = AIC(fit1)
fit2<-lm(y~x1+x2)
f2 = AIC(fit2)
fit3<-lm(y~x2+x3)
f3 = AIC(fit3)
fit4<-lm(y~x1+x3)
f4 = AIC(fit4)
      fit5<-lm(y~x1)
f5= AIC(fit5)
fit6<-lm(y~x2)
f6 = AIC(fit6)
fit7<-lm(y~x3)
f7 = AIC(fit7)
if(f1==min(c(f1,f2,f3,f4,f5,f6,f7))){
count <- count +1
}

}
print(sprintf("variance: %f beta: %f proportion: %f",
currentvariance,currentk,count/numberOfRuns))
```

```
}
}
```

B.4 Tables for chapter 4

Tables of simulation results for proportion of picking the true models

variance is 1, different sample sizes

	methods	proportion
$n = 5, \sigma^2 = 1$	AIC	0.997
	BIC	0.996
$n = 10, \sigma^2 = 1$	AIC	1
	BIC	1
$n = 50, \sigma^2 = 1$	AIC	1
	BIC	1
$n = 100, \sigma^2 = 1$	AIC	1
	BIC	1

variance is 5, different sample sizes

	methods	proportion
$n = 5, \sigma^2 = 5$	AIC	0.890
	BIC	0.909
$n = 10, \sigma^2 = 5$	AIC	0.995
	BIC	0.991
$n = 50, \sigma^2 = 5$	AIC	1
	BIC	1
$n = 100, \sigma^2 = 5$	AIC	1
	BIC	1

variance is 10, different sample sizes

	methods	proportion
$n = 5, \sigma^2 = 10$	AIC	0.747
	BIC	0.801
$n = 10, \sigma^2 = 10$	AIC	0.872
	BIC	0.859
$n = 50, \sigma^2 = 10$	AIC	1
	BIC	1
$n = 100, \sigma^2 = 10$	AIC	1
	BIC	1

variance is 20, different sample sizes

	methods	proportion
$n = 5, \sigma^2 = 20$	AIC	0.619
	BIC	0.649
$n = 10, \sigma^2 = 20$	AIC	0.577
	BIC	0.531
$n = 50, \sigma^2 = 20$	AIC	0.974
	BIC	0.924
$n = 100, \sigma^2 = 20$	AIC	0.998
	BIC	0.994

sample size is 5, different variances

	methods	proportion
$n = 5, \sigma^2 = 0.1$	AIC	1
	BIC	1
$n = 5, \sigma^2 = 1$	AIC	0.997
	BIC	0.996
$n = 5, \sigma^2 = 5$	AIC	0.890
	BIC	0.909
$n = 5, \sigma^2 = 10$	AIC	0.747
	BIC	0.801
$n = 5, \sigma^2 = 20$	AIC	0.619
	BIC	0.649

sample size is 10, different variances

	methods	proportion
$n = 10, \sigma^2 = 0.1$	AIC	1
	BIC	1
$n = 10, \sigma^2 = 1$	AIC	1
	BIC	1
$n = 10, \sigma^2 = 5$	AIC	0.995
	BIC	0.801
$n = 10, \sigma^2 = 10$	AIC	0.872
	BIC	0.859
$n = 10, \sigma^2 = 20$	AIC	0.577
	BIC	0.531

sample size is 50, different variances

	methods	proportion
$n = 50, \sigma^2 = 0.1$	AIC	1
	BIC	1
$n = 50, \sigma^2 = 1$	AIC	1
	BIC	1
$n = 50, \sigma^2 = 5$	AIC	1
	BIC	1
$n = 50, \sigma^2 = 10$	AIC	1
	BIC	1
$n = 50, \sigma^2 = 20$	AIC	0.974
	BIC	0.924

sample size is 100, different variances

	methods	proportion
$n = 100, \sigma^2 = 0.1$	AIC	1
	BIC	1
$n = 100, \sigma^2 = 1$	AIC	1
	BIC	1
$n = 100, \sigma^2 = 5$	AIC	1
	BIC	1
$n = 100, \sigma^2 = 10$	AIC	1
	BIC	1
$n = 100, \sigma^2 = 20$	AIC	0.998
	BIC	0.994

$n = 100$, different σ^2 and β

$(\beta_1, \beta_2, \beta_3)$	(0.2,1,1)		(1,5,5)		(2,10,10)	
variance	AIC	BIC	AIC	BIC	AIC	BIC
1	1	1	1	1	1	1
2	1	0.995	1	1	1	1
5	0.732	0.428	1	1	1	1
10	0.326	0.118	1	10.995	1	1
15	0.234	0.060	0.974	0.873	1	1
20	0.171	0.031	0.856	0.643	1	0.997
25	0.098	0.023	0.736	0.429	0.995	0.951

B.5 R code for chapter 5

```
#simulation for ARL in 100 times
library(MASS)
x<-list()
RL<-numeric(0)
for(j in 1:100){
xn<-c(numeric(0),numeric(0),numeric(0))
for(i in 1:10000){
x[[i]]<-mvrnorm(50,m,S)
xn<-cbind(xn,colMeans(x[[i]]))}
xn1<-t(xn)
m1<-t(m)
r<-1
n<-2
N=10001
Cn<-numeric(0)
Mn<-matrix(NA,N,3)
while(n <= N){
Mn[1,]<-m
```

```
Mn[n,]<-r*xn1[n-1,]+(1-r)*Mn[n-1,]
Cn[n]<-(Mn[n,]-m1)%*%solve(S)%*%t(Mn[n,]-m1)
n<-n+1
Cn<-c(Cn,Cn[n])
}
Cn<-Cn[2:10001]
Cn1<-cbind(Cn,c(1:10000))
a<-Cn1[Cn1[,1]>0.4,2]
RL[j]<-min(a)
RL<-c(RL,RL[j])
}
RL<-RL[1:100]
RL
hist(RL)
ARL<-median(RL)
ARL
```

Bibliography

[1] ACQUAH, H. D.-G. Comparison of Akaike information criterion (AIC) and Bayesian information criterion (BIC) in selection of an asymmetric price relationship. *Journal of Development and Agricultural Ecnomics 2(1)* (2010), 001–006.

[2] ANDERSON, D. *Model Based Inference in the Life Sciences: A Primer on Evidence*. Springer, 2008.

[3] BARNETT, V., AND LEWIS, T. *Outliers In Statistical Data,3rd Edition*. John Wiley and Sons Ltd, 1994.

[4] BEAVAN, R., AND LITCHFIELD, N. Vertical land movement around the new zealand coastline: implications for sea-level rise. Tech. Rep. ISSN 1177-2425, Institution of Geological and Nuclear Sciences Limited, 2012.

[5] BEN-GAL, I. *Outlier Detection*. Kluwer Academic Publishers, 2005.

[6] BLEWITT, G. *Basics of the GPS Technique: Observation Equations*. Swedish Land Survey, 1997.

[7] BURNHAM, K., AND ANDERSON, D. Multimodel inference understanding AIC and BIC in model selection. *Sociological Methods Research 33(2)* (2004), 261–304.

[8] CHANDOLA, V., BANERJEE, A., AND KUMAR, V. Outlier detection:a survey. Tech. rep., University of Minnesota.

[9] CHERNICK, M. R. *Bootstrap Methods: A Guide for Practitioners and Researchers*. Wiley Series in Probability and Statistics, 1999.

[10] D.M.HAWKINS, AND D.H.OLWELL. *Cumulative Sum Charts and Charting for Quality Improvement*. Springer, 1997.

[11] EFRON, B., AND TIBSHIRANI, R. J. *An Introduction to the Bootstrap*. Chapman and Hall, 1993.

[12] EI-RABBANY, A. *Introduction to GPS: the Global Positioning System*. Artech House mobile communications series, 2002.

[13] FOTHERINGHAM, A., BRUNSDON, C., AND CHARLTON, M. *Geographically Weighted Regression: the analysis of spatially varying relationships*. Wiley, 2003.

[14] FOX, D.AND HIGHTOWER, J., LIAO, L., AND SCHULZ, D.AND BORRIELLO, G. Bayesian filters for location estimation. Tech. Rep. 1536-1268, IEEE CS and IEEE ComSoc, 2003.

[15] FUJIKOSHI, Y., AND SATOH, K. Modified AIC and c_p in multivariate linear regression. *Biometrika 84* (1997), 707–716.

[16] HE, P. Some methods of deleting outliers from measuring data. *aviation metrology and measurement technology 14(1)* (1995).

[17] HODGE, V., AND AUSTIN, J. A survey of outlier detection methodologies. Tech. rep., Dept. of computer Science, University of York, York, 2004.

[18] HOFFMANN-WELLENHOF, B., LICHTENEGGER, H., AND COLLINS, J. *Global Positioning System: Theory and Practice, 3rd ed.* New York: Springer-Verlag, 1994.

[19] HURVICH, C., AND TSAI, C.-L. A CORRECTED AKAIKE INFORMATION CRITERION FOR VECTOR AUTOREGRESSIVE MODEL SELECTION. *Time series Analysis 14(3)* (2008), 271–279.

[20] JOHNSON, R. A., AND WICHERN, D. W. *Applied Multivariate Statistical Analysis, fifth Edition*. Pearson Education, 2002.

[21] KHAIRUNNIZA-BEJO, S., AND SHARIFF, A. Historical analysis of the land movement in landslide area using elastic image registration and conditional statement approach. *International Journal of Multimedia and Ubiquitous Engineering 6*, 3 (2011), 37–47.

[22] KHOO, M., WONG, V., ZHANG, W., AND CASTAGLIOLA, P. Optimal design of the synthetic chart for the process mean based on median run length. *IIE Transactions 44:9* (2012), 765–779.

[23] KITAGAWA, G. On the use of AIC for the detection of outliers. *Technometrics 21*, 2 (1979), 193–199.

[24] KLEUSBERG, A., AND LANGLEY, R. The limitations of GPS. *GPS World 1* (1990), 50–52.

[25] KOHNE, A., AND WOBNER, M. Transmitted GPS signals. "`http://www.kowoma.de/en/gps/index.htm`", 2009.

[26] KONISHI, S., AND KITAGAWA, G. *Information Criteria and Statistical Modeling*. Springer, Fukuoka and Tokyo, Japan, 2008.

[27] KOSTIUK, M. Using remote sensing data to detect sea level change. In *Land Satellite Information* (2002).

[28] LAI, T. Sequential changepoint detection in quality control and dynamical systems. *Journal of the Royal Statistical Society 57* (1995), 613–658.

[29] LIN, D.-B., AND JUANG, R.-T. Mobile location estimation based on differences of signal attenuations for GSM systems. Tech. Rep. NSC 92-2213-E-027-018, IEEE, 2003.

[30] LOWRY, C. A., WOODALL, W. H., CHAMP, C. W., AND RIGDON, S. E. A multivariate exponentially weighted moving average control chart. *Technometrics 34* (1992), 46–53.

[31] MANDEL, M., AND BETENSKY, R. A. Simultaneous confidence intervals based on the percentile bootstrap approach. *Comput Stat Data Anal. 52(4)* (2008), 2158–2165.

[32] MISRA, P., AND ENGE, P. *Global Positioning System:Signals, Measurements, and Performance*. Ganga-Jamuna Press, 2001.

[33] NATIONAL COORDINATION OFFICE FOR SPACE-BASED POSITIONING, N., AND TIMING. Selective availability. "http://www.gps.gov/systems/gps/modernization/sa/", 2012.

[34] NEWS, AND COMMUNICATIONS, R. New program to expand, enhance use of LIDAR sensing technology. "http://oregonstate.edu/ua/ncs/archives/2011/oct/new-program-expand-enhance-use-lidar-sensing-technology", 2011.

[35] PETOVELLO, M., AND O'DRISCOLL, C. Generating carrier phase measurements. "http://www.insidegnss.com/node/2146", 2010.

[36] PETTITT, A. A non-parametric approach to the change-point problem. *Journal of the Royal Statistical Society 28(2)* (1979), 126–135.

[37] POUSSEEUW, P., AND VAN ZOMEREN, B. Unmasking multivariate outliers and leverage points. *Journal of the American Statistical Association 85(411)* (1990), 633–639.

[38] RIZOS, C. Principles and practice of GPS surveying. "http://www.gmat.unsw.edu.au/snap/gps/gps_survey/principles_gps.htm", 1999.

[39] ROOS, T., MYLLYMAKI, P., AND TIRRI, H. A statistical modeling approach to location estimation. *IEEE Transactions on mobile computing 1*, 1 (2002), 59–69.

[40] SEGHOUANE, A.-K. New AIC corrected variants for multivariate linear regression model selection. *IEEE Transactions on aerospace and electronic systems 47* (2011), 1154–1164.

[41] SERVICE, N. C. GPS position accuracy measures. `http://www.novatel.com/assets/Documents/Bulletins/apn029.pdf`, 2003.

[42] SHAPIRO, S., AND WILK, M. Analysis of variance test for normality (complete samples). *Biometrika 52* (1965), 591–661.

[43] SHAW, M., SANDHOO, K., AND TURNER, D. Modernization of the global positioning system. *GPS World 11* (2000), 36–44.

[44] VILLASENOR ALVA, J. A., AND ESTRADA, E. G. A generalization of shapiro-wilk's test for multivariate normality. *Communications in Statistics-Theory and Methods 38:11* (2009), 1870–1883.

[45] ZAMBA, K., AND HAWKINS, D. A multivariate change-point model for change in mean vector and/or covariance structure. *Journal of quality Technology 41* (2009), 285–303.

[46] ZAMBA, K., AND HAWKINS, D. M. A multivariate change-point model for statistical process control. *Technometrics 48:4* (2006), 539–549.

[47] ZHAO, H., GAN, Z., AND XIAO, M. The method of judging outliers in multivariate statistical data. *Central China Normal University(Natural Science) 37(2)* (2003).

[48] ZHOU, J., NG, J.-Y., AND TONG, K. Using LDA method to provide mobile location estimation services within a cellular radio network. *Journal of computers 1*, 7 (2006), 41–50.